

98 P12 96

①9 BUNDESREPUBLIK  
DEUTSCHLAND



DEUTSCHES  
PATENTAMT

⑫ Übersetzung der  
europäischen Patentschrift

⑧7 EP 0 224 149 B1

⑩ DE 36 89 968 T 2

⑤1 Int. Cl. 5:  
**G 06 F 13/42**  
G 06 F 13/38  
H 04 L 7/00

- ②1 Deutsches Aktenzeichen: 36 89 968.2  
⑧6 Europäisches Aktenzeichen: 86 115 849.1  
⑧6 Europäischer Anmeldetag: 14. 11. 86  
⑧7 Erstveröffentlichung durch das EPA: 3. 6. 87  
④6 Veröffentlichungstag der Patentansprüche  
in deutscher Übersetzung: 25. 2. 88  
⑧7 Veröffentlichungstag  
der Patenterteilung beim EPA: 13. 7. 94  
④7 Veröffentlichungstag im Patentblatt: 1. 12. 94

③0 Unionspriorität: ③2 ③3 ③1

18.11.85 US 798822

⑦3 Patentinhaber:

Hayes Microcomputer Products, Inc., Norcross, Ga.,  
US

⑦4 Vertreter:

Solf, A., Dr.-Ing., 81543 München; Zapf, C., Dipl.-Ing.,  
Pat.-Anwälte, 42103 Wuppertal

⑧4 Benannte Vertragsstaaten:

BE, CH, DE, ES, FR, GB, IT, LI, NL, SE

⑦2 Erfinder:

Duncanson, Jay Paul, San Francisco, Calif. 94114,  
US; Murray, Mark Randall, Atlanta, George 30319,  
US

⑥4 Verbessertes synchrones/asynchrones Modem.

Anmerkung: Innerhalb von neun Monaten nach der Bekanntmachung des Hinweises auf die Erteilung des europäischen Patents kann jedermann beim Europäischen Patentamt gegen das erteilte europäische Patent Einspruch einlegen. Der Einspruch ist schriftlich einzureichen und zu begründen. Er gilt erst als eingelegt, wenn die Einspruchsgebühr entrichtet worden ist (Art. 99 (1) Europäisches Patentübereinkommen).

Die Übersetzung ist gemäß Artikel II § 3 Abs. 1 IntPatÜG 1991 vom Patentinhaber eingereicht worden. Sie wurde vom Deutschen Patentamt inhaltlich nicht geprüft.

DE 36 89 968 T 2

DE 36 89 968 T 2

Die vorliegende Erfindung betrifft ein Verfahren zum Austauschen von Informationsdaten zwischen Vorrichtungen, indem die Informationsdaten zu einem Informationsdatenwort codiert werden, welches eine erste vorbestimmte Parität besitzt, und die Informationsdaten als eine Serie von Datenworten übertragen werden, welche Serie eine Vielzahl derartiger Informationsdatenworte umfaßt, sowie zum Austauschen von Steuerdaten zwischen den Vorrichtungen.

Die Veröffentlichung Mini-Micro Conference Record, Mai 1984 Northeast, N.Y., USA von S.J.Durham mit dem Titel "Intelligent 212A-Type Modem Component" beschreibt einen über einen Universalasynchron- bzw. Universalsynchron-Sendeempfänger mit einem System-Mikrocomputerbus verbindbaren Datensendeempfänger bzw. Modem zum Senden und Empfangen von Daten in einem asynchronseriellen Datenformat.

Ferner beschreibt das Dokument EP-A-O 142 302 einen Modem, der Steuer- und Dateninformation von den Busleitungen eines Computers empfängt und als Antwort darauf, Daten über Fernsprechleitungen in einem asynchronseriellen Datenformat sendet und empfängt.

In der Fachzeitschrift ELECTRONIC DESIGN, Band 31, Nr. 8, April 1983 beschreibt R. Allan in seinem Beitrag mit dem Titel "Data-Comm, Telecomm ICS-Moving to Single Chips and More Intelligence" einen Modem, der in Verbindung mit dem Paralleldatenbus eines Mikrocomputers steht, und welcher Daten in entweder einem synchronseriellen Datenformat oder in einem asynchronseriellen Datenformat senden und empfangen kann.

Ein Modem (Modulator-Demodulator) ist eine Vorrichtung, die über Sprachfrequenz-Fernsprechleitungen Kommunikation zwischen Digitalcomputern erlaubt. Zwar verwenden viele Digitalkommunikationsvorschriften Asynchrondatenübertragung, dennoch gibt es viele Computer, die durch die Verwendung von Synchrondatenübertragung kommunizieren. Synchrondatenübertragung

stellt an sich kein Problem dar, da Vorrichtungen für sowohl Synchron- als auch Asynchron-Übertragung im Handel erhältlich sind.

In Asynchronkommunikation wird ein Zeichen durch die serielle Übertragung einer Anzahl von Bits nach einem bestimmten Zeichenformat gesendet. In einem typischen Zeichneformat wird das erste Bit als Startbit bezeichnet und es signalisiert dem Empfänger, daß eine Datenübertragung folgt. Die sieben darauffolgenden Bits werden Datenbits genannt und stellen das zu übertragende Zeichen dar. Das darauffolgende Bit ist häufig ein Paritätsbit, das zur Fehlerkontrolle der Datenbits verwendet wird. Das Schlußbit nennt man Stoppbit. Dieses signalisiert dem Empfänger, daß die Datenübertragung damit beendet ist. Somit enthält jedes Telegramm ein Zeichen. Diese Bitfolge wird dann für das nächste Zeichen sowie für alle darauffolgenden Zeichen wiederholt, bis alle Zeichen übertragen worden sind. Ein Zeichen kann zum Beispiel eine Buchstabe, eine Ziffer, eine Interpunktion oder Steuerinformationen sein.

Verschiedene Zeichenformate werden verwendet. Die Anzahl der Datenbits kann typischerweise 5, 6, 7 oder 8 betragen, wobei das Paritätsbit geradzahlig, ungeradzahlig oder Nullparität bedeuten kann, es kann aber auch gestrichen werden, und die Anzahl der Stoppbits kann 1,  $1\frac{1}{2}$  oder 2 betragen. Darüber hinaus die typische Geschwindigkeit, mit der die Daten gesendet werden, entspricht einer der folgenden Angaben: 110, 150, 300, 600, 1200, 2400, 4800, 9600, oder 19.200 Bits pro Sekunde (bps).

Damit zwei Vorrichtungen kommunizieren können, müssen beide das gleiche Zeichenformat sowie die gleiche Datenübertragungsgeschwindigkeit in Bits pro Sekunde verwenden.

Bei einer Synchronkommunikation kann ein Telegramm statt lediglich ein Zeichen viele Zeichen enthalten. In einem typischen Synchron-Telegrammformat signalisieren die ersten acht Bits, im allgemein als Anfangskennzeichen bezeichnet, den Anfang eines Telegramms. Die darauffolgenden acht Bits sind Adressenbits. Diese geben den Zielort des Telegramms bekannt. Die darauffolgende Bitreihe stellt die zu übertragenden Zeichen

bzw. Informationen dar. Diese Reihe kann von beliebiger Länge sein und wird praktisch nur durch die Anzahl von Bits begrenzt, die normalerweise empfangsmäßig zu erwarten sind, bevor ein Fehler auftritt. Die darauffolgenden 16 Bits sind eine Folge von Bits zur Kontrolle des Datenübertragungsblocks. Diese Bits werden zur Fehlererkennung benutzt. Die letzten acht Bits signalisieren das Ende des Telegramms und werden allgemein als Ende-Flag bezeichnet. Somit kann jedes Telegramm viele Zeichen beinhalten.

Wie bei Asynchronkommunikation gibt es mehrere verschiedene Telegrammformate und Datenübertragungsgeschwindigkeiten im allgemein auch bei Synchronkommunikation, und damit zwei Vorrichtungen kommunizieren können, müssen sie beide das gleiche Telegrammformat und die gleiche Datenübertragungsgeschwindigkeit benutzen.

Manche Computer verfügen über eine Schlitzöffnung für eine Kommunikationskarte. In diese Schlitzöffnung kann der Benutzer entweder eine Karte für eine serielle Kommunikation einschieben, wobei die Karte dann mit einem Modemtreiber verbunden wird, oder er kann eine Karte einschieben, die einen kompletten Modem enthält. Es ist typisch, daß solche Karten nur in Verbindung mit Asynchronkommunikation kompatibel sind, und sie beinhalten ein Asynchron-Kommunikationslement, wie etwa das von der Fa. National Semiconductor Corporation, Santa Clara, California hergestelltes Element INS8250A, sowie weitere für das Senden and Empfangen von Signalen über eine Fernsprechleitung erforderliche Komponenten. Die Kommunikationsschnittstelle zwischen einem Computer und einer auf einer Karte in Verbindung mit dem Schlitz stehenden Vorrichtung ist normalerweise asynchron.

Es ist möglich, eine mit sowohl Synchron- als auch Asynchronkommunikation kompatible Modemkarte herzustellen, indem das Element INS8250A durch einen Controller für kombinierte Synchron/Asynchronkommunikation wie z.B. durch das von der Fa. Zilog, Inc., Cambell, California hergestellte Element Z8530 ersetzt wird. Derartige Controller kosten jedoch im allgemein mehr als die Asynchronkommunikationslemente. Daher müssen Modem-Hersteller unter drei verschiedenen Möglichkeiten mit

unterschiedlichen Vor- und Nachteilen hinsichtlich Kosten und Leistung wählen.

Entschließt sich ein Modem-Hersteller, einen sowohl synchron- als auch asynchronkompatiblen Modem herzustellen, muß er die Herstellungskosten drücken oder seine Gewinnmarge schneiden, um wettbewerbsfähig zu bleiben. Sonst würden potentielle Kunden, die lediglich eine asynchrone Kompatibilität benötigen, bei einem anderen Hersteller bestellen, der nur asynchrone Modems zu einem günstigeren Preis verkauft. Ferner kann eine Erweiterung um die synchrone Kompatibilität den Modem mit einigen bereits bestehenden Asynchron-Datenübertragungsprogrammen inkompatibel gestalten.

Stellt der Modem-Hersteller lediglich Asynchronmodems her, läuft er Gefahr, potentielle Kunden zu verlieren, die sowohl eine synchrone als auch eine asynchrone Lösung benötigen.

Stellt der Modem-Hersteller wiederum zwei Modem-Ausführungen her, eine synchron, und eine asynchron, so belastet er sich mit den Kosten und Problemen zweier Produktionsstraßen statt nur eine betreiben zu müssen.

Gleichermaßen stehen Modem-Anwender vor der Wahl unter drei Ausführungen wählen zu müssen: einen Nur-Asynchronmodem zu kaufen, und dabei auf Synchronkommunikation zu verzichten; einen Nur-Synchronmodem zu kaufen, und dabei auf Asynchronkommunikation zu verzichten;; oder einen Synchron-/Asynchronmodem zu kaufen, und dafür einen höheren Preis zu bezahlen.

Es besteht daher der Bedarf für einen Modem, der ein Asynchronkommunikationselement verwendet, das sowohl Synchron- als auch Asynchron-Kommunikation erlaubt, das in der Preisklasse von Nur-Asynchronmodems liegt und das mit bereits bestehenden asynchronen Datenkommunikationsprogrammen kompatibel ist.

Darüber hinaus ist es bekannt, daß bei synchronen Kommunikationsformaten die Daten mit einer Geschwindigkeit bereitgestellt werden müssen, die dazu ausreicht, um sicherzustellen, daß der volle Dateninhalt eines wie oben beschriebenen kompletten Telegramms zur Verfügung steht. Auch wünschenswert ist es, eine Normschnittstelle für eine Synchron-

/Asynchronkommunikation bereitzustellen, die es dem Anwender ermöglicht, Kommunikation synchron oder asynchron einfach durchzuführen, indem er einigen einfachen Regeln befolgt, und zwar auf eine Weise, die von der jeweiligen Modemvorrichtung unabhängig ist.

Daher ist es eine Aufgabe der vorliegenden Erfindung es einem Modem zu ermöglichen, Kommunikationen sowohl synchron als auch asynchron durchzuführen, und dies bei geringeren Kosten als bei einem Modem, das ein Synchron-/Asynchronkommunikationselement benützt.

Gemäß der vorliegenden Erfindung wird diese Aufgabe durch ein Verfahren nach dem einzigen Anspruch gelöst. Somit werden Befehlsanweisungen durch den Einsatz eines invertierten Paritätsbits codiert.

Auf ein in einem ausgehenden Datensignal enthaltenes erstes Befehl zur Lieferung eines ersten seriellen Datensignals spricht ein Datenwandler an, indem er das abgehende Datensignal in einem seriellen Asynchrondatenformat beläßt, und ein ankommendes Datensignal liefert, indem er ein zweites serielles Datensignal in dem seriellen Asynchrondatenformat beläßt, und auf ein in dem ausgehenden Datensignal enthaltenes zweites Befehl zur Lieferung des ersten seriellen Datensignals spricht er an, indem er das abgehende Datensignal aus dem seriellen Asynchrondatenformat in ein serielle Synchrondatenformat wandelt und das ankommende Datensignal liefert, indem er das zweite serielle Datensignal aus dem seriellen Synchrondatenformat in das serielle Asynchrondatenformat wandelt. Der Datenwandler liefert das ankommende Datensignal in dem seriellen Asynchrondatenformat mit einem ersten Paritätsbit und fügt Steuerdaten in das ankommende Datensignal ein, indem er ein Datenwort mit einem invertierten Paritätsbit sendet, so daß der Datenwandler sowohl das zweite serielle Datensignal als auch die Steuerdaten als ein ankommendes Datensignal liefert, mit Hilfe dessen die Steuerdaten erfindungsgemäß codiert werden. Eine am Datenwandler

angeschlossene Vorrichtung prüft die Parität eines jeden in dem seriellen Asynchrondatenformat enthaltenen Wortes und, auf Grund der Parität eines jeden Datenwortes, bestimmt, ob es sich bei einem Wort um zweite serielle Daten oder um Steuerdaten handelt.

Grob ausgedrückt, liefert die vorliegende Erfindung ein Verfahren, mit dem ein Modem sowohl synchron als auch asynchron kommunizieren kann und mit bestehenden Asynchron-Datenkommunikationsprogrammen kompatibel bleibt, wobei es nur jene Komponenten benötigt, die für Asynchronkommunikation benötigt werden.

Feiner ausgedrückt, liefert die vorliegende Erfindung ein Verfahren, mit dem ein Modem mit einem Parallelbus-Ein-/Ausgangs-Port (E/A-Port) und einem seriellen E/A-Port über den seriellen E/A-Port sowohl synchron als auch asynchron kommunizieren kann.

Modemsteuerbefehle können an einem ersten Punkt in einen Datenstrom einer Synchronkommunikation eingefügt und an einem zweiten Punkt aus dem Datenstrom entfernt werden, so daß der ursprüngliche Datenstrom erhalten bleibt. Ferner lassen sich Modemparameter-Statuswörter an einem ersten Punkt in den Datenstrom einer Synchronkommunikation einfügen und an einem zweiten Punkt aus dem Datenstrom entfernen, so daß der ursprüngliche Datenstrom erhalten bleibt.

Nähere Einzelheiten der vorliegenden Erfindung ergeben sich aus der folgenden Beschreibung einer bevorzugten Ausführungsform. In der Zeichnung zeigen:

- Fig. 1 eine schematische Darstellung der bevorzugten Ausführungsform der vorliegenden Erfindung.
- Fig. 2 eine Darstellung des HDLC/SDLC-Datenformats
- Fig. 3 eine Darstellung des BSC-Datenformats

Fig. 4 ein Zustandsdiagramm für den Sender der bevorzugten Ausführungsform

Fig. 5 ein Zustandsdiagramm für den Empfänger der bevorzugten Ausführungsform.

In der Zeichnung, in deren Figuren gleiche Komponenten durch gleiche Kennziffern gekennzeichnet sind, ist Fig. 1 eine schematische Darstellung der bevorzugten Ausführungsform der vorliegenden Erfindung.

Externe Vorrichtung 9 ist eine Rechnervorrichtung wie ein Computer, Prozessor oder Datenterminal. Die bevorzugte Ausführung der vorliegenden Erfindung ist zur Verwendung mit einem IBM-Personalcomputer auslegt. Daher ist externe Vorrichtung 9 im Sinne der bevorzugten Ausführungsform als ein IBM-PC zu verstehen. Dem Fachmann sind Einzelheiten der Busstruktur und des zeitlichen Signalverlaufs des IBM-PCs wohl bekannt. Diese gehen aus der Druckschrift "IBM PC Technical Reference Manual" von April 1984 hervor, auf die hiermit Bezug genommen wird. Externe Vorrichtung 9 verfügt über einen Ein-/Ausgangsbuss (E/A-Bus) 10, der sowohl Daten als auch Befehle trägt. Datenbus 11 des E/A-Bus 10 besteht aus acht Leitungen für die Übertragung eines 8-Bit-Parallelwortes. Datenbus 11 ist mit E/A-Port A eines bidirektionellen Tristate-Pufferspeichers 12 verbunden. E/A-Port B des Pufferspeichers 12 ist durch 8-Bit-Datenbus 13 mit den Leitungen D0 bis D7 des E/A-Ports des Universal-Asynchronempfängersenders (UART) 18 verbunden. UART 18 wird auch als Asynchron-Kommunikationselement (AKE) allgemein genannt. In der bevorzugten Ausführungsform ist UART 18 das von der Fa. National Semiconductor Corporation, Santa Clara, California hergestellte Element INS8250A. Dem Fachmann sind Einzelheiten der Betriebsweise und des Innenaufbaus von UART 18 wohl bekannt. Diese gehen aus Druckschriften des Herstellers, der Fa. National Semiconductor Corporation hervor, und werden daher in der folgenden Beschreibung nicht näher erläutert.



Adressenbus 14 ist mit den Eingängen eines Adressendecoders 16 verbunden. Leitung 15 zur Signalfreigabe ist mit dem Freigabeeingang des Adressendecoders 16 verbunden. Der Ausgang des Adressendecoders 16 ist durch Leitung 17 mit dem NICHT-Freigabeeingang des Pufferspeichers 12 verbunden, der die Ausgänge des Pufferspeichers 12 freigibt, sowie mit dem NICHT-Eingang der Chipwahl 2 (CS2) von UART 18 verbunden. Die Eingänge der Chipwahl 0 und 1 (CS0, CS1) von UART 18 sind durch Leitung 38 mit logisch 1 verbunden.

Adressbus 20 ist mit den Adresseneingängen A0-A2 von UART 18 verbunden. Hierdurch kann die externe Vorrichtung 9 das Register von UART 18 anwählen, um Daten einzuschreiben bzw. auszulesen. Leitung 22 zum Abfragen der Eingangsdaten (DISTR) ist mit dem NICHT-Eingang für die abgefragten Eingangsdaten von UART 18 verbunden. Leitung 21 zum Abfragen der Ausgangsdaten (DOSTR) ist mit dem NICHT-Ausgang für die abgefragten Ausgangsdaten von UART 18 verbunden. Der NICHT-Eingang der abgefragten Adresse (ADS), der Eingang der abgefragten normalen Eingangsdaten, der Eingang der abgefragten normalen Ausgangsdaten und der NICHT-Eingang Datensatz bereit von UART 18 liegen an logisch 0 durch Leitung 19.

Der Fachmann versteht, daß Datenbus 11, Adressenbusse 14 und 20, Adressen-Freigabeleitung 15 sowie die Datenabfrage auf Leitungen 21 und 22 auf gewöhnliche Weise eingesetzt werden, um Daten zwischen der externen Vorrichtung 9 und UART 18 zu übertragen.

Der Fachmann versteht ohnehin bzw. durch Lesen der Druckschriften des Herstellers von UART 18, daß UART 18 über elf zugängliche Register verfügt. Externe Vorrichtung 9 sendet Befehle an UART 18 dadurch, daß sie die Daten in diese Register schreibt, und das Status von UART 18 durch Lesen der in diesen Registern enthaltenen Daten liest.

Der Treibersperrausgang (DDIS) von UART 18 ist durch Leitung 25 mit dem Datenrichtungseingang (DIR) des Pufferspeichers 12 und einem ersten Eingang der Interruptlogikschaltung 24 verbunden. UART 18 belegt Leitung 25 mit logisch 0, wenn Daten aus Datenport D0-D7 von UART 18

ausgelesen werden. Logisch 0 auf Leitung 25 teilt Pufferspeicher 12 mit, daß Port B ist Eingang und Port A ist Ausgang. Logisch 1 auf Leitung 25 erlaubt eine Datenübertragung von UART 18 an externe Vorrichtung 9, und aktiviert die Interruptlogik 24. Aus den Herstellerangaben geht hervor, daß Ausgang DDIS von UART 18 logisch 0 ist, wenn Daten aus UART 18 an Datenbus 13 ausgelesen werden.

Der zweite NICHT-Ausgang (OUT2) und der Interruptausgang (INT) von UART 18 sind durch Leitungen 27 bzw. 26 mit einem zweiten bzw. dritten Eingang der Tristate-Interruptlogik 24 verbunden. Belegt UART 18 Leitung 27 mit logisch 1, ist der Ausgangswiderstand der Interruptlogik 24 hoch. Belegt UART 18 Leitung 25 mit logisch 1 und Leitung 27 mit logisch 0, entspricht der Ausgang der Interruptlogik 24 dem Interruptausgang von UART 18. Der Ausgang der Interruptlogik 24 ist durch Leitung 23 mit Bus 10 verbunden. Das durch UART 18 auf Leitung 27 gesetztes Signal OUT2 wird von externer Vorrichtung 9 durch einen Befehl an UART 18 gesteuert. Somit kann externe Vorrichtung 9 die Interruptfähigkeiten von UART 18 freigeben oder sperren.

Rücksetzleitung 30 von Bus 10 ist mit dem Rücksetzeingang von UART 18 und mit einem ersten Eingang von Rücksetzlogik 31 verbunden. Der erste NICHT-Ausgang (OUT1) von UART 18 ist durch Leitung 32 mit einem zweiten Eingang der Rücksetzlogik 31 verbunden. Der Ausgang der Rücksetzlogik 31 ist durch Leitung 33 mit dem NICHT-Rücksetzeingang des Prozessors 47 sowie mit dem NICHT-Rücksetzeingang der Decodierlogik und den Latchspeichern 77 verbunden. Logisch 1 auf Leitung 30 bewirkt ein Rücksetzen von UART 18, Prozessor 47 sowie Decodierlogik 77.

Darüber hinaus kann externe Vorrichtung 9 UART 18 befehlen, Leitung 32 mit logisch 0 zu belegen, und Prozessor 47 und Decodierlogik 77 zurückzusetzen. Rücksetzlogik 31 liefert einen Einzelausgangsimpuls von 10 Mikrosekunden Pulsdauer dann, wenn Leitung 30 mit logisch 0 bzw. Leitung 32 mit logisch 0 belegt ist. Aufbaumöglichkeiten der Rücksetzlogik 31 sind dem Fachmann wohl bekannt.

Ein erster Ausgang des Taktgebers 42 ist durch Leitung 41 mit dem Eingang XTAL1 von UART 18 verbunden. In der bevorzugten Ausführungsform beträgt die Frequenz dieses ersten Ausgangs 1,843 MHz. Ein zweiter Ausgang des Taktgebers 42 ist durch Leitung 43 mit dem Eingang XTAL1 des Prozessors 47 verbunden. Das Komplement dieses zweiten Ausgangs ist durch Leitung 44 mit dem Eingang XTAL2 des Prozessors 47 verbunden. In der bevorzugten Ausführungsform beträgt die Frequenz dieses zweiten Ausgangs 7,372 MHz.

Prozessor 47 ist ein Mikroprozessor wie z.B. das von der Fa. Zilog, Inc., Campbell, California hergestellte Element Z8681. Prozessor 47 verfügt über vier 8-bit Ports, nämlich Port 0 (P00-P07), Port 1 (P10-P17), Port 2 (P20-P27) und Port 3 (P30-37). Einzelheiten der Betriebsweise des Elements Z8681 sind vom Hersteller erhältlich.

Der asynchronserielle Datenausgang (SOUT) von UART 18 ist durch Leitung 35 mit dem Eingang P30 des Prozessors 47 und mit dem Eingang Y0 eines Dual-Vier-zu-Eins-Multiplexers 56 verbunden. Der Ausgang P20 des Prozessors 47 ist durch Leitung 50 mit den Eingängen Y1 und Y2 des Multiplexers 56 verbunden. Der Eingang Y3 des Multiplexers 56 ist durch Leitung 51 mit logisch 1 belegt. Der Ausgang Y des Multiplexers 56 ist durch Leitung 57 mit dem Sendedateneingang (CTXD) des Modemtreibers 60 verbunden.

Es ist ohne weiteres ersichtlich, daß die dem Eingang CTXD des Modemtreibers 60 bereitgestellten Daten als die SOUT-Daten von UART 18, die P20-Daten vom Prozessor 47 oder als logisch 1 auf Leitung 51 gewählt werden können. Durch diese Wahl und Streckenführung der zu sendenden Daten lassen sich Format und Protokoll für die Datenausgabe sowohl synchron als auch asynchron entsprechend wählen.

Der Empfangsdatenausgang (CRXD) des Modemtreibers 60 ist durch Leitung 55 mit dem Eingang P27 des Prozessors 47 sowie mit den Eingängen X0 und X2 des Multiplexers 56 verbunden. Der Ausgang P37 des Prozessors 47 ist durch Leitung 52 mit den Eingängen X1 und X3 des Multiplexers 56 verbunden. Der Ausgang X

des Multiplexers 56 ist durch Leitung 34 mit dem Eingang für asynchronserielle Daten (SIN) von UART 18 verbunden.

Es ist ferner ohne weiteres erischlich, daß die vom Modemtreiber 60 ausgegebenen Empfangsdaten selektiv geführt werden können, so daß sie direkt zu dem Eingang SIN von UART 18 oder indirekt über Prozessor 47 geführt werden. Durch diese Wahl und Streckenführung der Empfangsdaten lassen sich Format und Protokoll für die Dateneingabe sowohl synchron als auch asynchron entsprechend wählen.

Modemtreiber 60 enthält Modulator, Demodulator, Taktgeber, Logik und Register zur Betriebssteuerung und Speicherung der Betriebsstatusdaten. Dem Fachmann sind die Aufbaumöglichkeiten des Modemtreibers 60 wohl bekannt. Insbesondere läßt sich Modemtreiber 60 durch die Implementierung eines bzw. mehrerer Chipsätze realisieren, die im Handel erhältlich sind bzw. mit MSI- und LSI-ICs aufgebaut werden können.

Modemtreiber 60 verfügt über zwei Taktgeberausgänge: den Taktgeber für die Sendedaten (CTXCLK) und den Taktgeber für die Empfangsdaten (CRYCLK). Ausgang CTXCLK ist durch Leitung 64 mit dem Eingang P31 des Prozessors 47 verbunden und dient dazu, die Übertragung der Sendedaten vom Prozessor 46 zum Modemtreiber 60 zu synchronisieren. Ausgang CRXCLK ist durch Leitung 65 mit dem Eingang P32 des Prozessors 47 verbunden und dient dazu, die Übertragung der Empfangsdaten vom Modemtreiber 60 zum Prozessor 46 zu synchronisieren. Der Ausgang CTXCLK und Ausgang CRYCLK werden nur im Synchronbetrieb verwendet.

Ausgänge P21 und P22 des Prozessors 47 sind durch Leitungen 53 bzw. 54 mit den Steuereingängen A bzw. B des Multiplexers 56 verbunden. Somit steuert der Prozessor 47 den Betrieb des Multiplexers 56, je nachdem ob Synchron- oder Asynchronbetrieb gewünscht ist.

Der Ausgang P26 des Prozessors 47 ist durch Leitung 66 mit dem NICHT-Rücksetzeingang des Modemtreibers 60 verbunden. Der Lese-/Nichtschreib-Ausgang (R/W) des Prozessors 47 ist durch Leitung 67 mit einem ersten Eingang der Lese-/Schreiblogikschaltung 71 verbunden. Der NICHT-Ausgang der Datenabfrage (DS) des Prozessors 47 ist durch Leitung 70 mit

einem zweiten Eingang der Lese-/Schreiblogik 71 sowie mit dem NICHT-Eingang der Datenabfrage (DS) der Decodierlogik 77 verbunden. Der Lese-Ausgang der Lese-/Schreiblogik 71 ist durch Leitung 72 mit dem Lese-Eingang des Modemtreibers 60 verbunden. Der NICHT-Schreibausgang der Lese-/Schreiblogik 71 ist durch Leitung 73 mit dem NICHT-Schreibeingang des Modemtreibers 60 verbunden. Lese-/Schreiblogik 71 wandelt das Einzel-Lese-/Schreibsignal auf Leitung 67 in zwei getrennte Signale auf Leitungen 72 und 73 um. Lese-/Schreiblogik 71 wird durch logisch 0 auf Leitung 70 freigegeben.

Ein-/Ausgänge P10-P17 des Prozessors 47 sind durch den 8-Bit-Zweirichtungsdatenbus 74 mit den Ein-/Ausgängen D0-D7 des Modemtreibers 60, den Adressen-/Datenbuseingängen (ADB) der Decodierlogik 77 sowie mit den Ausgängen D0-D7 des ROM-Speichers 90 verbunden. Modemtreiber 60 verfügt über Betriebsstausregister und Betriebssteuerregister. Lese-/Schreiblogik 71, Decodierlogik 77 sowie Datenbus 74 ermöglichen das Auslesen/Einschreiben der Register im Modemtreiber 60 und damit den Betrieb des Modemtreibers 60 zu überwachen und zu steuern.

Ausgänge P00-P07 des Prozessors 47 sind durch den 8-Bit-Adressenbus 75 mit den Adressenbus-Eingängen (AB) der Decodierlogik 77 sowie mit den Eingängen (A8-A12) der fünf höchstwertigen Bitadressen des ROMs 90 verbunden. Der NICHT-Ausgang für die Adressenabfrage (AS) des Prozessors 47 ist durch Leitung 76 mit dem NICHT-Eingang für die Adressenabfrage der Decodierlogik 77 verbunden. Decodierlogik 77 nimmt die auf dem Adressen-/Datenbus 74 und dem Adressenbus 75 bestehenden Signale an, um mehrere Ausgangssignale zu liefern.

Der erste Steuerausgang (C1) der Decodierlogik 77 ist durch eine sechsadrige Leitung 87 mit den sechs Steuereingängen des Modemtreibers 60 verbunden. In der bevorzugten Ausführungsform werden diese sechs Steuereingänge zum Anwählen und Freigeben eines bestimmten Registers im Modemtreiber 60 für Lese- bzw. Schreiboperationen benützt.

Der zweite Steuerausgang (C2) der Decodierlogik ist durch eine vieradrige Leitung 86 mit den vier Steuereingängen der Fernsprechnschnittstelle 80 verbunden, um den (nicht

dargestellten) Gabelumschalteranschluß der Fernsprechleitung 81 zu steuern, und um eine weitere (nicht dargestellte) Vorrichtung mit der Fernsprechleitung 81 zu verbinden bzw. von dieser zu trennen, usw.

Der Ausgang (TXD) für die modulierten Sendedaten des Modemtreibers 60 ist durch Leitung 82 mit dem Eingang (TXD) für die modulierten Sendedaten der Leitungsschnittstelle 80 verbunden. Der Ausgang (RXD) für die modulierten Empfangsdaten der Leitungsschnittstelle 80 ist durch Leitung 85 mit dem Eingang (TXD) für die modulierten Empfangsdaten des Modemtreibers 60 verbunden. Die Fernsprech-Endgeräte (TIP und RING) der Leitungsschnittstelle 80 sind mit der Fernsprechleitung 81 verbunden. Da Fernsprechleitung 81 Mittel zur seriellen Datenübertragung in beiden Richtungen darstellt, enthält Leitungsschnittstelle 80 eine (nicht dargestellte) Sende-Empfangs-Weiche für die Belegung der Fernsprechleitung mit dem TXD-Signal und für den Abruf des RXD-Signals von der Fernsprechleitung 81.

Modemtreiber 60 enthält einen Zweiton-Mehrfrequenz-(DTMF)-Tongenerator für die Tonwahl der Teilnehmerrufnummern. Der Ausgang DTMF des Modemtreibers 60 ist durch Leitung 83 mit dem Eingang DTMF der Leitungsschnittstelle 80 verbunden. Leitungsschnittstelle 80 belegt die Fernsprechleitung 81 mit diesen DTMF-Signalen. Modemtreiber 60 beinhaltet auch einen Impulsgenerator für die Impulswahl der Teilnehmerrufnummern. Der Impulsausgang des Modemtreibers 60 ist durch Leitung 84 mit dem Impulseingang der Leitungsschnittstelle 90 verbunden. Leitungsschnittstelle 80 belegt die Fernsprechleitung 81 ebenfalls mit diesen Impulswahlsignalen. Dem Fachmann sind Aufbaumöglichkeiten der Leitungsschnittstelle 80 wohl bekannt.

Fernsprechleitung 81 ist mit dem Fernsprech-Endgeräten der entfernten Vorrichtung 100 verbunden. Entfernte Vorrichtung 100 ist ein typisches Datenendgerät mit einem Synchron- und/oder Asynchronmodem. Es ist einleuchtend, daß die vorliegende Erfindung den Zweck hat, externe Vorrichtung 9 mit entfernter Vorrichtung 100 entweder synchron oder asynchron kommunizieren zu lassen.

Decodierlogik 77 erlaubt es dem Prozessor 47 auch Daten vom ROM 90 zu lesen. Die Adressenausgänge (A0-A7) der Decodierlogik sind durch den ROM-Adressenbus 91 mit den unteren 8-Bit-Eingängen (A0-A7) des ROM 90 verbunden. Der NICHT-Ausgang für die Chipwahl der Decodierlogik 77 ist durch Leitung 92 mit dem NICHT-Eingang für die Chipwahl des ROM 90 verbunden. Der NICHT-Ausgang der Ausgangsaktivierung (OE) der Decodierlogik 77 ist durch Leitung 93 mit dem NICHT-Eingang der Ausgangsaktivierung des ROM 90 verbunden. Decodierlogik 77 liefert die entsprechenden Signale an Bus 90 und Leitungen 92 und 93, welche zusammen mit den Signalen auf Bus 75 es dem Prozessor ermöglichen, Daten von ROM 90 über Bus 74 zu lesen. Dem Fachmann sind Aufbaumöglichkeiten der Decodierlogik 77 zur Steuerung des Auslesens der Daten aus ROMs wohl bekannt. Es ist einleuchtend, daß ROM 90 Betriebsanweisungen und Datankonstanten für Prozessor 47 beinhaltet.

Der NICHT-Ausgang der Sendebereitschaft (CTS) der Decodierlogik 77 ist durch Leitung 45 mit dem NICHT-Eingang CTS von UART 18 verbunden. Das NICHT-CTS-Signal wird im Synchronbetrieb für die Steuerung des Datenstroms von UART 18 zum Prozessor 47 benützt, Ist Prozessor 74 bereit, zusätzliche Daten von UART 18 (über SOUT) anzunehmen, belegt Prozessor 47 Bus 74 und Bus 75 sowie Leitungen 70 und 76 mit Signalen, so daß Decodierlogik 77 die NICHT-CTS-Leitung 45 mit logisch 0 belegt.

Der NICHT-Ausgang der Datenträgererfassung (DCD) der Decodierlogik 77 ist durch Leitung 46 mit dem NICHT-DCD-Eingang von UART 18 verbunden. Das Vorhandensein eines Datenträgers von der entfernten Vorrichtung 100 auf der Fernsprechleitung 81 wird vom Modemtreiber 60 erfaßt, und veranlaßt den Modemtreiber 60 das Erkennungszeichen der Datenträgererfassung (Flag DCD) in eines seiner Statusregister zu setzen. Prozessor 47 liest die Statusregister des Modemtreibers 60 regelmäßig. Ist das Flag DCD in Modemtreiber 60 gesetzt, veranlaßt Prozessor 47 die Decodierlogik 77 Leitung 46 mit logisch 0 zu belegen. Ein logisch 0 auf Leitung 46 setzt das Flag DCD in das Statusregister von UART 18. Bei jedem Setzen bzw. Rücksetzen eines DCD-Flags in UART 18 erzeugt UART 18 ein Interrupt.

Externe Vorrichtung 9 bedient sich des Interrupt, liest das Statusregister von UART 18 und erfaßt dadurch die Änderung beim Datenträger auf Leitung 81.

Es wird nun vorausgesetzt, daß Synchronbetrieb der Kommunikation mit 2400 bps gewählt worden ist. Daten werden von der entfernten Vorrichtung 100 durch den Modemtreiber 60 mit 2400 Bits pro Sekunde empfangen. Die demodulierten Daten werden nun mit dieser Übertragungsgeschwindigkeit vom Multiplexer 56 zum Prozessor 47 weitergleitet. Prozessor 47 wird nun die empfangenen Daten-, Additionsstart-, Stopp- und Paritätsbits decodieren und die empfangenen Daten über Multiplexer 56 an den SIN-Eingang von UART 18 asynchron senden. Dabei werden die synchron empfangenen Daten vom Modemtreiber 60 an den Prozessor 47 mit einer Übertragungsgeschwindigkeit von 2400 bps. gesendet. Prozessor 47 muß die erforderliche Löschung des Nullbits und die Decodierung NRZ/NTZ1 der ankommenden Daten vornehmen, Start-, Stopp- und Paritätsbits zu den verarbeiteten Daten hinzufügen, und die Daten an UART 18 asynchron senden. Ferner, wie weiter unten erläutert, kann es notwendig sein, daß Prozessor 47 ein oder mehrere Statusworte in den Datenstrom an UART 18 einfügt. Weil Prozessor 47 mehr Bits an UART 18 senden muß als Prozessor 47 vom Modemtreiber 60 empfängt, muß die Datenübertragungsgeschwindigkeit zwischen Prozessor 47 und UART 18 größer sein als die Datenübertragungsgeschwindigkeit zwischen Modemtreiber 60 und Prozessor 47. In der bevorzugten Ausführungsform wurde eine Datenübertragungsgeschwindigkeit zwischen Prozessor 47 und UART 18 von 9600 bps gewählt.

Prozessor 47 muß warten, bis Modemtreiber 60 ein komplettes Datenwort empfangen hat, bevor Prozessor 47 die für eine Asynchronübertragung notwendigen zusätzlichen Bits an UART 18 hinzufügen kann. Dies bedeutet, daß die empfangenen Daten etwa 1 Millisekunde nachdem die modulierten Daten vom Modemtreiber 80 empfangen wurden, in UART 18 bereitstehen. UART 18 erzeugt ein Interrupt, wenn Prozessor 47 ein Datenwort empfangen hat. Externe Vorrichtung 9 muß dann dieses Interrupt abarbeiten, um das neue Datenwort von UART 18 lesen zu können. Es ist also erforderlich, daß externe Vorrichtung 9 das Interrupt von UART



18 schnell abarbeitet und die Daten in UART einliest. Übermäßige Verzögerungen beim Abarbeiten des Interrupts oder beim Lesen der Daten von UART 18 können zu Problemen führen, wenn im BSC-Betrieb Prozessor 47 in den Flagsuchzustand zurückversetzt wird, wie unten erläutert, wenn in den Betriebsarten HDLC/SDLC ein Abbruchsignal gesendet wird, ebenfalls unten erläutert, oder bereits empfangene Daten können verlorengehen.

Auf gleiche Weise sendet UART 18 Sendedaten an Prozessor 47 asynchron mit 9600 bps. Prozessor 47 muß die Start-, Stopp- und Paritätsbits löschen, die Daten codieren, und die Daten zum Modemtreiber 60 mit 2400 bps senden. Prozessor 47 hat eine beschränkte Speicherkapazität und, weil Daten mit 9600 bps von UART 18 empfangen und an den Modemtreiber mit 2400 bps gesendet werden, benötigt Prozessor 47 Mittel zum Unterbrechen des Datenstroms von UART 18. Die Leitung CTS (45) wird für Handshaking zwischen UART 18 und Prozessor 47 verwendet, so daß der effektive Datendurchsatz zum Prozessor 47 von UART 18 etwa dem Datendurchsatz vom Prozessor 47 zum Modemtreiber 60 entspricht.

Der NICHT-Ausgang für die Bereitschaft des Datenendgerätes (DTR) von UART 18 ist durch Leitung 36 mit dem Eingang P33 des Prozessors 47 verbunden. Der NICHT-Ausgang für Sendebereitschaft (RTS) von UART 18 ist durch Leitung 37 mit dem Eingang P25 des Prozessors 47 verbunden. Der Ausgang P23 des Prozessors 47 ist durch Leitung 40 mit dem NICHT-Eingang der Rufanzeige (RI) von UART 18 verbunden. Die Verwendung dieser Signale ist unten erläutert.

Der NICHT-Baudausgang (BAUD) von UART 18 ist durch Leitung 39 mit dem Empfängertakteingang (RCLK) von UART 18 verbunden. Diese Verbindung veranlaßt UART 18, die gleiche Übertragungsgeschwindigkeit für das Senden und Empfangen der Daten wie Prozessor 47 zu verwenden.

#### Weitere Realisierungsmöglichkeiten

Fig. 1 zeigt die bevorzugte Ausführungsform in Verbindung mit dem Paralleldatenbus 10 einer externen Vorrichtung 9. Es ist jedoch einleuchtend, daß in manchen Anwendungen externe

Vorrichtung 9 jene Komponenten beinhalten kann, die links von der gestrichelten Trennlinie 99 gezeigt sind und welche über eine genormte Schnittstelle RS-232 (Leitungen 34, 35, 36, 37, 40, 45 und 46) verfügen. Bei dieser Anwendung, da wahrscheinlich keine Rücksetzleitung 33 zur Verfügung steht, wird Software verwendet, um Prozessor 47 das Rücksetzbefehl zu geben, entweder intern oder durch Erweiterung der Decodierlogik 77 durch einen adressierbaren Rücksetzausgang. Darüber hinaus muß die externe Verbundvorrichtung 9 programmiert sein, um die unterschiedlichen Bedeutungen der Signale auf einigen dieser Leitungen zu erkennen. Die vorliegende Erfindung ist also auf eine Verwendung mit externen Vorrichtungen 9 mit einem zugänglichen Paralleldatenbus nicht beschränkt.

### Programmierung

In der bevorzugten Ausführungsform die Bedeutung einiger der Eingangssignale und der Ausgangssignale von UART 18 ist nicht die genormte RS-232-Bedeutung. Daher wird externe Vorrichtung 9 programmiert, um die folgenden Bedeutungen dieser Signale zu erkennen. Im Synchronbetrieb verarbeitet Prozessor 47 sowohl Sende- als auch Empfangsdaten. Er kann also so aufgefaßt werden, daß er einen Prozessor für seriell eingegebene/seriell ausgegebene Sendedaten und einen Prozessor für seriell eingegebene/seriell ausgegebene Empfangsdaten beinhaltet, die einfach als Sender bzw. Empfänger bezeichnet werden.

Das Signal DTR auf Leitung 36 gibt Prozessor 47 die Anweisung, Synchronbetrieb zu verlassen und Asynchronbetrieb aufzunehmen. Wenn Prozessor 47 von Synchronbetrieb in Asynchronbetrieb umschaltet, muß Prozessor 47 Modemtreiber 60 und Leitungsschnittstelle 800 zu erkennen geben, ob die Verbindung zu entfernter Vorrichtung 100 über Fernsprechleitung 81 aufrechterhalten oder abgebrochen werden soll. Daher muß externe Vorrichtung 9 Prozessor 47 bereits angewiesen haben, diese Verbindung nach Wunsch des Benützers aufrechtzuerhalten oder abzubrechen.

Das Signal RTS auf Leitung 37 gibt Prozessor 47 die Anweisung, den Empfänger im Prozessor 47 zurückzusetzen (den

Empfänger zwingen, den Flagsuchzustand aufzunehmen). Das Signal RI auf Leitung 40 teilt externer Vorrichtung 9 mit, daß ein Blockende gesendet wird und daß der Sender im Prozessor 47 den Ruhezustand aufnehmen wird.

Das Signal CTS auf Leitung 45 teilt externer Vorrichtung 9 mit, daß Prozessor 47 zu diesem Zeitpunkt nicht in der Lage ist, weitere Sendedaten anzunehmen. Das Signal DCD auf Leitung 46 teilt externer Vorrichtung 9 mit, daß Modemtreiber 60 einen Datenträger auf Fernsprechleitung 81 erfaßt hat.

Externe Vorrichtung 9 programmiert UART 18, bei einem Zustandswechsel der Signale RI, CTS und DCD ein Interrupt zu erzeugen. Externe Vorrichtung 9 muß auch Prozessor 47 hinsichtlich der Programmierung des Modemtreibers 60 in Kenntnis setzen.

Wie weiter unten erläutert wird, sendet Prozessor 47 gelegentlich Statusworte an externe Vorrichtung 9 über UART 18. In der bevorzugten Ausführungsform beinhalten die Statusworte folgende Informationen: Blockende empfangen und Blockprüffolge korrekt; Blockende empfangen und Blockprüffolge inkorrekt; Leer-Flag empfangen; Leerzeichen empfangen; Synchronbetrieb verlassen; Synchronbetrieb aufnehmen; RTS-Befehl quittiert (Empfänger von Prozessor 47 wurde zurückgesetzt); und Empfangsdaten enthält Abbruchsignal.

#### **Asynchrondatenbetrieb - Allgemein**

Im Asynchrondatenbetrieb belegt Prozessor 47 Leitungen 53 und 54 mit logisch 0. Dies veranlaßt Multiplexer 56 seinen Eingang Y0 mit seinem Ausgang Y, und seinen Eingang X0 mit seinem Ausgang X zu verbinden. Dies verbindet den Sendedatenausgang (SOUT) von UART 18 mit dem Sendedateneingang (CTXD) des Modemtreibers 60, und verbindet den Empfangsdatenausgang (CRXD) des Modemtreibers 60 mit dem Empfangsdateneingang (SIN) von UART 18. Hierdurch werden die Sende- und Empfangsdaten zwischen UART 18 und Modemtreiber 60 übertragen, ohne daß sie durch Prozessor 47 beeinflußt werden. Die Datenübertragung zwischen UART 18 und Modemtreiber 60 ist asynchron und der auf Leitungen 64 bzw. 65 vom Modemtreiber 60

gelieferte Sendedatentakt (CTXCLK) bzw. Empfangsdantentakt (CRXCLK) werden von UART 18 nicht benützt.

Sendedaten von UART 18 auf Leitung 35 und Empfangsdaten vom Modemtreiber 60 auf Leitung 55 werden auf Umschaltfolgen, Betriebsanweisungen, usw. vom Prozessor 47 überwacht.

### **Asynchronbefehlsbetrieb - Allgemein**

Externe Vorrichtung 9 gibt Prozessor 47 die Anweisung, die Asynchronbefehlsbetrieb aufzunehmen, indem sie UART 18 veranlaßt, die DTR-Leitung 36 mit logisch 0 zu belegen oder durch einen über den SOUT-Ausgang von UART 18 gesendeten Befehl. Prozessor 47 wird auch bei Erstinbetriebnahme Asynchronbefehlsbetrieb aufnehmen. Hat Prozessor 47 Asynchronbefehlsbetrieb aufgenommen, programmiert externe Vorrichtung 9 den Prozessor 47 durch Schreiben der entsprechenden Daten in die Betriebssteuerregister des Prozessors 47. Dies wird von externer Vorrichtung 9 dadurch erzielt, daß sie die entsprechenden Befehlsworte in den Sendepufferregister von UART 18 schreibt. Asynchron hierzu sendet dann UART 18 diese Befehlsworte zum Prozessor 47. Da sich Prozessor 47 im Asynchronbefehlsbetrieb befindet, deutet Prozessor 47 diese Worte als Befehle und reagiert entsprechend. Dieses Verfahren wird auch verwendet, damit externe Vorrichtung 9 dem Prozessor 47 befehlen kann, Modemtreiber 60 mit den gewünschten Betriebsparametern zu programmieren.

### **Synchronbetrieb - Allgemein**

Im Synchronbetrieb belegt Prozessor 47 Leitung 53 mit logisch 0 und Leitung 54 mit logisch 1. Dementsprechend verbindet Multiplexer 56 seinen Eingang Y1 mit seinem Ausgang Y und seinen Eingang X1 mit seinem Ausgang X.

Sendedaten fließen von UART 18 über Leitung 35 zum Prozessor 47, werden vom Prozessor 47 verarbeitet, und fließen dann vom Prozessor 47 über Leitung 50, Multiplexer 56 und Leitung 57 zum Modemtreiber 60. Empfangsdaten fließen vom Modemtreiber 60 über Leitung 55 zum Prozessor 47, werden vom

Prozessor 47 verarbeitet, und fließen dann vom Prozessor 47 über Leitung 52, Multiplexer 56 und Leitung 34 zu UART 18.

Im Synchronbetrieb fließen Daten asynchron zwischen UART 18 und Prozessor 47, und synchron zwischen Prozessor 47 und Modemtreiber 60.

Ist der Sendepuffer in UART 18 leer, d.h. das letzte Sendewort in UART 18 wurde an Prozessor 47 gesendet, erzeugt UART 18 ein Interrupt. Dies veranlaßt externe Vorrichtung 9 das Interrupt zu lesen und, über Bus 10, das nächste Sendewort an UART 18 zu senden. Die Steuerung des Sendedatenstroms von externer Vorrichtung 9 zum Prozessor 47 erfolgt durch Verwendung des Eingangs CTS von UART 18 auf Leitung 45 asynchron. Bei jedem Zustandswechsel des Signals CTS auf Leitung 45 durch Prozessor 47, erzeugt UART 18 ein Interrupt. Externe Vorrichtung 9 arbeitet das Interrupt ab und stellt den Zustand des Signals CTS auf Leitung 45 fest. Ist Prozessor 47 bereit mehr Daten zu empfangen, belegt er Leitung 45 mit logisch 0. Dies teilt externer Vorrichtung 9 mit, das nächste Sendezeichenwort über UART 18 an Prozessor 47 zu senden. Kann Prozessor 47 keine weitere Daten annehmen, belegt er Leitung 45 mit logisch 1, wodurch externer Vorrichtung 9 mitgeteilt wird, keine weiteren Worte über UART 18 an Prozessor 47 zu senden. Das Signal CTS auf Leitung 45 hat keinen Einfluß auf den SOUT-Sender von UART 18. Es ist einleuchtend, daß zwischen dem Zeitpunkt zu dem Prozessor 47 das Signal CTS umschaltet und dem Zeitpunkt zu dem externe Vorrichtung 9 das Interrupt abarbeitet und entsprechend reagiert eine endliche Verzögerung stattfindet.

Hätte Prozessor 47 einen Sendepuffer mit lediglich einem Wort, könnte Prozessor 47 wegen dieser Verzögerung keine weitere Sendedaten haben und fälschlicherweise mit dem Senden einer Blockende beginnen, oder Sendedaten könnten wegen des Überschreibens des Sendepuffers verlorengehen, da UART 18 etwaige in ihrem Senderhalteregister enthaltenen Sendedaten automatisch sendet. Um das Auftreten dieser unerwünschten Bedingungen zu vermeiden, wird Prozessor 47 so programmiert, daß er über einen Vierwort-Sendesilospeicher verfügt. Der Vierwort-Sendesilospeicher gleicht die Verzögerung aus, so daß die

Wahrscheinlichkeit eines Eintretens einer fehlerhaften Blockende-Sendung oder des Überschreibens von Sendedatenworte erheblich reduziert wird.

Prozessor 47 veranlaßt Decodierlogik 77 CTS-Leitung 45 mit logisch 0 zu belegen, wenn noch ein oder zwei Wörter im Sendesilospeicher übrig geblieben sind. und CTS-Leitung 45 mit logisch 1 zu behaften, wenn der Sendesilospeicher mehr als zwei Wörter beinhaltet. Es ist zu beachten, daß sollte der Sendesilospeicher des Prozessors 47 leer werden, eine Blockendenfolge von Prozessor 47 an Modemtreiber 60 gesendet wird.

UART 18 belegt Leitung 26 mit einem Interruptsignal (logisch 1), wenn ein Zustandswechsel des NICHT-Signals CTS erfolgt. Dies veranlaßt Interruptlogik 24 Leitung 23 von Bus 10 mit einem Interruptsignal zu belegen. Erkennt externe Vorrichtung 9 ein Interruptsignal auf Leitung 23, verlangt sie folglich ein Interruptstatuswort (liest die Inhalte des Interrupt-Kennzeichnungsregisters) von UART 18. Folglich überwacht externe Vorrichtung 9 den Status des NICHT-Signals CTS durch UART 18.

Wird das NICHT-Signal CTS auf Leitung 45 zu logisch 0, darf externe Vorrichtung 9 nicht zu lange verzögern, bevor sie das nächste Sendezeichen an UART 18 sendet, weil wie bereits erläutert, Prozessor 47 ein Blockende an Modemtreiber 60 sendet, wenn der Sendesilospeicher leer wird. Wird der Sendesilospeicher leer, wird Prozessor 47 auch die NICHT-Leitung der Rufanzeige (RI) 40 mit logisch 1 belegen. Auch dies veranlaßt UART 18 ein Interruptsignal auf Leitung 23 zu erzeugen. Externe Vorrichtung 9 infolge des Interruptsignals verlangt das Interruptstatuswort von UART 18. Somit kann externe Vorrichtung 9 das Senden eines Blockendes feststellen.

Stellt externe Vorrichtung 9 fest, daß ein Blockende fehlerhaft gesendet wird, d.h. das zu sendende Telegramm unvollkommen gesendet wurde, dann belegt externe Vorrichtung 9 Bus 10 mit einer Anweisung (BREAK) an UART 18, welche UART 18 veranlaßt, ein Abbruchzeichen an Prozessor 47 zu senden. Prozessor 47, der noch mit dem Senden der Blockendenfolge an

Modemtreiber 60 beschäftigt ist, bricht daraufhin die Blockendensendung ab und sendet anschließend das Abbruchzeichen an den Modemtreiber 60. Das Abbruchzeichen teilt entfernter Vorrichtung 100 mit, daß das an sie gesendete Telegramm einen Fehler beinhaltet. Externe Vorrichtung 9 sollte so konfiguriert sein, daß sie die erneute Sendung des Telegramms veranlaßt.

Ein synchroner Sendedatenstrom vom Prozessor 47 zum Modemtreiber 60 wird durch Verwendung des Taktausgangs der Sendedaten (CTXCLK) des Modemtreibers 60 auf Leitung 64 ausgeführt. Jeder Taktimpuls auf Leitung 64 veranlaßt Prozessor 47, Leitung 50 mit einem Bit des im Sendesilospeicher enthaltenen Sendeworts zu belegen. Leitung 50 ist mit dem Eingang für serielle Daten (CTXD) des Modemtreibers 60 verbunden.

Sind sämtliche Bits in einem im Sendesilospeicher gepufferten Sendewort gesendet worden, beginnt Prozessor 47 mit dem Senden der Bits des darauffolgenden im Sendesilospeicher gepufferten Sendeworts. Sind sämtliche im Sendesilospeicher gepufferten Sendeworte zum Modemtreiber 60 ausgetaktet worden, und sind keine neue Worte empfangen worden, so daß der Sendesilospeicher leer ist, beginnt Prozessor 47 mit dem Senden eines Blockendes an Modemtreiber 60.

Ein synchroner Empfangsdatenstrom vom Modemtreiber 60 zum Prozessor 47 wird durch Verwendung des Taktausgangs der Empfangsdaten (CRXCLK) des Modemtreibers 60 auf Leitung 65 ausgeführt. Jeder Taktimpuls auf Leitung 65 taktet ein Bit des Empfangsdatenwortes in Prozessor 47. Prozessor 47 verfügt ebenfalls über einen Vierwort-Empfangssilospeicher. Sind sämtliche Bits in einem Empfangswort vom Prozessor 47 empfangen worden, belegt Prozessor 47 den Empfangssilospeicher mit dem Wort. Anschließend beginnt Prozessor 47 mit der Zusammenstellung des nächsten Empfangswortes.

Der Empfangsdatenstrom vom Prozessor 47 zu UART 18 erfolgt asynchron. Sobald der Prozessor 47 ein Empfangswort zusammengestellt und es durch den Empfangssilospeicher bewegt hat, sendet Prozessor 47 das Wort im Asynchronformat über Leitung 52, Multiplexer 56 und Leitung 34 an UART 18.

Hat UART 18 ein komplettes Wort vom Prozessor 47 empfangen, erzeugt UART 18 ein Interrupt, das externe Vorrichtung 9 veranlaßt, das Interrupt abzuarbeiten und das empfangene Wort zu lesen.

### Synchronformate

Prozessor 47 verarbeitet die Sendeworte und Empfangsworte nach dem gewählten Synchronformat. Eine Vielzahl von Synchrondatenformaten sind entwickelt worden. HDLC-, SDLC- und binärsynchrone (BSC) Datenübertragungsformate wurden für die bevorzugte Ausführungsform gewählt, weil diese Formate am breitesten Verwendung finden. Es versteht sich jedoch, daß andere Synchrondatenformate ohne weiteres implementiert werden können.

Der Blockaufbau für HDLC- und SDLC-Betrieb ist in Fig. 2 gezeigt. Das HDLC/SDLC-Erkennungszeichen (Flag) 101 hat den Binärwert 01111110. Das Adressenzeichen 102 ist die Adresse der Zielstation des Telegramms.

Im Empfangsdatenbetrieb kann Prozessor 47 durch externe Vorrichtung 9 angewiesen werden, das Adressenzeichen 102 zu prüfen oder zu ignorieren. Erhält Prozessor 47 die Anweisung, das Adressenzeichen 102 zu prüfen, sendet Prozessor 47 keinen Teil des Telegrammes an externe Vorrichtung 9, wenn das Adressenzeichen 102 weder der Stationsadresse des Anwenders noch der Rundrufadresse (binär 11111111) entspricht. Entspricht das Adressenzeichen 102 entweder der Stationsadresse des Anwenders oder der Rundrufadresse bzw. sollte Prozessor 47 die Anweisung erhalten, Adressenzeichen 102 zu ignorieren, dann behandelt Prozessor 47 Adressenzeichen 102 als Daten und sendet das gesamte Telegramm an externe Vorrichtung 9.

Im Empfangsbetrieb werden die N Datenbits 103 und die Blockprüffolge (FCS) vom Prozessor 47 verarbeitet und an externe Vorrichtung 9 gesendet. Da N eine unbestimmte Variable ist, kann Prozessor 47 den Unterschied zwischen Daten 103 und FCS 104 erst dann erkennen, wenn das Schlußflag 105 empfangen wird. Flag 105 ist mit Flag 101 identisch. Definitionsgemäß ist FCS die 16 Bits, die Flag 105 unmittelbar einleiten. Wenn also Prozessor 47



Flag 105 erfaßt, führt Prozessor 47 das Senden der Daten 103 und FCS 104 zur externen Vorrichtung zu Ende durch, vergleicht die Ergebnisse seiner FCS-Berechnungen für die empfangenen Daten 103 mit FCS 104, und sendet ein Statuswort an externe Vorrichtung 9. Das Statuswort teilt externer Vorrichtung 9 mit, daß ein Blockende empfangen wurde und ob der FCS-Vergleich zu einem korrekten oder inkorrekten Ergebnis geführt hat.

Im Sendebetrieb erzeugt externe Vorrichtung 9 Adressenzeichen 102 und Prozessor 47 verarbeitet Adressenzeichen 102 genau wie Daten 103.

Im Sendebetrieb erzeugt Prozessor 47 FCS-104 und Flag 105, sobald der Sendesilospeicher leer wird. Wie bereits erläutert, wenn der Sendesilospeicher leer wird, meldet Prozessor 47 auch externer Vorrichtung 9, daß ein Blockende gesendet wird. Es ist einleuchtend, daß eine Blockendenfolge aus FCS-104 und Flag 105 besteht. Sollte also Prozessor 47 mit der Erzeugung einer Blockendenfolge beginnen, bevor das Telegramm komplett ist, muß externe Vorrichtung 9 das Abbruchsignal an Prozessor 47 senden, bevor Flag 105 komplett übermittelt worden ist.

Der Blockaufbau des BSC-Formats ist in Fig. 3 gezeigt. Die Synchronisierungszeichen 110 und 111 sind stets identisch und sind programmierbar. Externe Vorrichtung 9 teilt Prozessor 47 den Wert der Synchronisierungszeichen 110 und 111 mit. Jedes der N Datenzeichen 112a-112n hat eine Länge von 8 Bits.

Im Sendebetrieb, wenn Prozessor 47 ein erstes Datenzeichen 112a von externer Vorrichtung 9 empfängt, erzeugt und sendet Prozessor 47 Synchronisierungszeichen 110 und 111 an Modemtreiber 60. Anschließend beginnt Prozessor 47 mit dem Senden der Daten 112 an Modemtreiber 60. Wird der Sendesilospeicher leer, erzeugt und sendet Prozessor 47 je nach Wahl der Steuerung der externen Vorrichtung 9 Mark-Ruhe- bzw. Synchronisierungszeichen.

Im BSC-Betrieb teilt Prozessor 47 externer Vorrichtung 9 über Leitung 40 und UART 18 mit, daß der Sendesilospeicher leer ist, ohne jedoch eine Blockendenfolge (CRC (Cyclic Redundancy Check) 113 und PAD 114) zu erzeugen. Externe Vorrichtung 9 erzeugt das Blockende und sendet es an Prozessor 47. Darauf hin

sendet Prozessor 47 die Blockendenfolge an Modemtreiber 60, als ob es sich dabei um Daten handeln würde. Pad 114 hat den Binärwert 11111111.

Im Empfangsbetrieb startet Prozessor 47 im Flagsuchzustand. Daher sendet Prozessor 47 zunächst keine Empfangsdaten an externe Vorrichtung 9, bis er Synchronisierungszeichen 110 und 111 erfaßt hat. Prozessor 47 streift Synchronisierungszeichen 110 und 111 von dem empfangenen Telegramm ab und sendet das restliche empfangene Telegramm an externe Vorrichtung 9.

Obwohl es möglich ist, erfaßt Prozessor 47 keine Blockendenfolge wegen Einschränkungen in der Zeit und ROM 90 Speicherplatzmangel. Daher sendet Prozessor 47 an die externe Vorrichtung 9 CRC 113, Pad 114 sowie alle Teile der nachfolgenden Blöcke einschl. Synchronisierungszeichen 110 und 111, bis externe Vorrichtung 9 dem Prozessor 47 die Anweisung gibt, den Flagsuchzustand erneut aufzunehmen.

Im Empfangsbetrieb bleibt Prozessor 47 im Flagsuchzustand und sendet das Empfangswort an die externe Vorrichtung erst dann, wenn Flag 101 empfangen worden ist. Prozessor 47 nimmt den Flagsuchzustand erneut auf, nachdem eine Abbruchfolge empfangen wurde oder, wenn externe Vorrichtung 9 Prozessor 47 die Anweisung gibt, den Flagsuchzustand erneut aufzunehmen.

In der bevorzugten Ausführungsform gibt externe Vorrichtung 9 Prozessor 47 die Anweisung, den Flagsuchzustand erneut aufzunehmen, indem sie über Bus 10 eine Anweisung an UART 18 sendet, welche UART 18 veranlaßt, NICHT-Leitung 37 einer Sendeanforderung (RTS) mit logisch 0 zu belegen. Diese Anweisung den Flagsuchzustand erneut aufzunehmen gilt für die Formate HDLC, SDLC und BSC. Prozessor 47 quittiert die Flagsuchanweisung durch Senden eines Statusworts an externe Vorrichtung 9 über UART 18.

Befindet sich Prozessor 47 im Flagsuchzustand, bleibt er im Flagsuchzustand und sendet keine Empfangsdaten an externe Vorrichtung 9, bis Prozessor 47 die beiden Synchronisierungszeichen 111 und 112 erfaßt, die anschließend Anderes als ein Marke/Ruhe-Zeichen aufweisen.

Folglich muß externe Vorrichtung 9 den einkommenden Datenstrom prüfen, um CRC 113 und Pad 114 in den Daten zu erfassen und davon zu trennen, und außerdem die Anweisung an Prozessor 47 geben, den Flagsuchzustand erneut aufzunehmen.

Es ist einleuchtend, daß da die Kommunikation zwischen Prozessor 47 und UART 18 asynchron erfolgt, wird UART 18 ein Sendewort auf Bus 10 mit Start-, Stopp- und Paritätsbits ergänzen, bevor das Sendewort an Prozessor 47 gesendet wird. Diese zusätzlichen Bits müssen für eine synchrone Übertragung vom Sendewort entfernt werden. Daher entfernt Prozessor 47 die Start-, Stopp- und Paritätsbits, bevor er ein Sendewort an Modemtreiber 60 sendet.

Auf gleiche Weise im Empfangsbetrieb fügt Prozessor 47 die Start-, Stopp- und Paritätsbits zu jedem vom Modemtreiber 60 empfangenen Wort hinzu, bevor er das Empfangswort an UART 18 sendet. UART 18 löscht dann die Start-, Stopp- und Paritätsbits, bevor sie Bus 10 mit dem empfangenen Wort belegt.

In den Betriebsarten HDLC und SDLC führt Prozessor 47 auch das Einfügen des Nullbits in Sendeworte und das Löschen des Nullbits von Empfangsworten durch. Es ist einleuchtend, daß das Einfügen/Löschen von Nullbits im HDLC- und SDLC-Betrieb erforderlich ist, um zu verhindern, daß eine Kombination der Zeichen Adressen 102 und/oder Daten 103 den Binärwert eines Flags 101 oder 105 hat. Zu diesem Zweck zählt Prozessor 47 die Anzahl der aufeinanderfolgenden logisch 1 Vorkommnisse in dem von UART 18 gesendeten Datenwort. Hat Prozessor 47 fünf aufeinanderfolgende logisch 1 Vorkommnisse gezählt, fügt er ein logisch 0 in das an Modemtreiber gesendete Wort ein. Dementsprechend löscht Prozessor 47 ein etwaige nach fünf aufeinanderfolgende logisch 1 Vorkommnisse auftretendes logisch 0 von den vom Modemtreiber 60 empfangenen Daten.

Bei Flags 101 und 105 oder bei einer Abbruchfolge (Binärwert 1111 1111) werden keine Nullen eingefügt oder gelöscht. Erfasst Prozessor 47 eine Abbruchfolge im Empfangswort, veranlaßt er UART 18 ein Interrupt an externe Vorrichtung 9 zu senden und darüber hinaus ignoriert er den Rest des durch die Abbruchfolge unterbrochenen Blocks.

Beim BSC-Betrieb ist ein Einfügen/Löschen von Nullbits nicht erforderlich.

Prozessor 47 erhält die Anweisung von externer Vorrichtung 9, ob die Codierung nach dem NTZ-Verfahren oder nach dem invertierten NTZ-Verfahren (NRZI) erfolgen soll. Dementsprechend führt der Prozessor 47 die Codierung des Sendeworts nach dem NTZ- oder dem NTZI-Verfahren durch, bevor er es an Modemtreiber 60 sendet, und führt die Decodierung des Empfangsworts nach dem NTZ- oder dem NTZI-Verfahren durch, bevor er es an UART 18 sendet.

### Zustandsdiagramme

Fig. 1 ist ein Zustandsdiagramm des Senders HDLC/SDLC. Der Sender HDLC/SDLC verfügt über fünf Zustände: Marke/Ruhe 120, Flag 121, Daten 122, FCS 123, und Abbruch 124. Ein Sender-Rücksetzbefehl 119 veranlaßt den Sender, den Marke/Ruhe-Zustand 120 aufzunehmen.

Der Sender wählt entweder den Marke/Ruhe-Zustand 120 oder den Flag-Zustand 121, wenn der Sender ruht, d.h. wenn keine Daten zum Senden vorhanden sind. Im Marke/Ruhe-Zustand 120 belegt Ausgang P20 mit logisch 1. Ein Abbruchbefehl (BREAK SIGNAL) von externer Vorrichtung 9 hat keinen Einfluß auf den Sender in diesem Zustand. Der Sender erhöht seinen Zustand auf Flag-Zustand 121, wenn er Sendedaten von UART 18 erhält.

Im Flag-Zustand 121 erzeugt und sendet der Sender das Flagzeichen HDLC/SDLC kontinuierlich. Der Sender kann über UART 18 von externer Vorrichtung 9 den Befehl erhalten, den Abbruchzustand 124 aufzunehmen. Nachdem der Abbruchbefehl verarbeitet und das Abbruchzeichen gesendet wurde, sendet der Sender erneut Flagzeichen. Der Sender erhöht seinen Zustand auf Datenzustand 122, sobald er Sendedaten von UART 18 erhält. Hat er aus dem Marke/Ruhe-Zustand auf Grund des Erhalts von Sendedaten von UART 18 den Flag-Zustand 121 angenommen, dann bleibt der Sender im Flag-Zustand 121 nur solange, um das Eröffnungsflag zu senden, bevor er seinen Zustand auf Datenzustand 122 erhöht.

Im Datenzustand 122 nimmt der Sender Sendedaten von UART 18 kontinuierlich an, führt bei Bedarf das Hinzufügen eines Nullbits aus, belegt Ausgang P20 mit den Daten, und aktualisiert die Blockprüffolge (FCS). Hat der Sender keine weiteren Sendedaten, erhöht er seinen Zustand auf FCS-Zustand 123. Sendet die externe Vorrichtung 9 den Abbruchbefehl, erhöht der Sender seinen Zustand auf Abbruchzustand 124.

Im FCS-Zustand 123 sichert der Sender die Blockprüffolge (FCS), führt bei Bedarf das Hinzufügen eines Nullbits aus und belegt Ausgang P20 mit den FCS-Daten. Nachdem die FCS-Daten gesendet worden sind, erhöht der Sender seinen Zustand auf Flag-Zustand 121, um das Schlußflag zu senden. Wurde Flag-Ruhezustand 121 gewählt, setzt der Sender das Senden des Flag-Ruhezeichens fort. Wurde Marke-Ruhezustand 120 gewählt, erhöht der Sender seinen Zustand auf Marke-Ruhezustand 120, nachdem das Schlußflag gesendet wurde.

Abbruchzustand 124 kann aus jedem Zustand, auch aus der erneuten Aufnahme des Abbruchzustands 124, aufgenommen werden. Im Abbruchzustand 124 sendet der Sender eine Abbruchfolge (acht aufeinanderfolgende logisch 1 Bits), und dann erhöht er seinen Zustand auf entweder Marke/Ruhe-Zustand 120 oder Flag-Ruhezustand 121, je nachdem wie gewählt wurde. Erhält der Sender von externer Vorrichtung 9 den Befehl, Abbruchzustand 124 aufzunehmen, erhöht der Sender sofort seinen Zustand auf Abbruchzustand 124 und fängt mit dem Senden des Abbruchzeichens an, ohne auf das Senden oder Fertigsenden etwaiger von UART 18 empfangenen Daten, des FCS-Zeichens, oder etwaiger Flag-Zeichen zu warten.

Der Sender BSC ist dem Sender HDLC/SDLC gleich, mit Ausnahme: (1) kein Zustand 123; (2) tritt ein Unterlauf des Sendesilospeichers ein, sendet der Sender BSC das Sendesilospeicher-Leersignal und anschließend erhöht er seinen Zustand auf Zustand 120 oder Zustand 121, je nachdem wie gewählt wurde; und (3) im Zustand 121 sendet der Sender BSC die BSC-Synchronisierungszeichen 110, 111.

Fig. 5 ist ein Zustandsdiagramm des Empfängers HDLC/SDLC. Auch der Empfänger HDLC/SDLC verfügt über fünf Zustände:

Flagsuchen nach Flag 130, Flagsynchronisierung 131, Adresse 132, Daten 133, und Blockende 134. Ein Empfänger-Rücksetzbefehl 129 veranlaßt den Empfänger, den Flagsuchzustand 130 aufzunehmen.

Im Flagsuchzustand 130 tastet der Empfänger die vom Modemtreiber 60 am Eingang P27 ankommenden Daten auf die Suche nach einem Flagzeichen ab. Ist ein Flagzeichen erfaßt, erhöht der Empfänger seinen Zustand auf Flagsynchronisierungszustand 131.

Im Flagsynchronisierungszustand 131 hat der Empfänger ein oder mehrere Flagzeichen erfaßt und Zeichensynchronisation erreicht. Der Empfänger verbleibt in diesem Zustand, bis ein Nichtflagzeichen erfaßt ist. Enthält das Nichtflagzeichen sieben oder mehr aufeinanderfolgende Bits logisch 1, wie in dem Marke/Ruhe-Zeichen, kehrt der Empfänger in den Flagsuchzustand 130 zurück. Enthält das Nichtflagzeichen weniger als sieben aufeinanderfolgende Bits logisch 1, geht der Empfänger davon aus, daß es sich dabei um das erste Zeichen des Adressenfelds 102 handelt, und der Empfänger nimmt den Adressenzustand 132 auf.

Nach Aufnahme des Adressenzustands 132 setzt der Empfänger den Empfangs-FCS-Generator vorab auf alle logisch 1 Bits. Wurde der Empfänger für Adressenerkennung programmiert, vergleicht der Empfänger das Empfangsadressenzeichen mit der programmierten Adresse; und außerdem mit der "Rundruf"-Adresse (Binärwert 11111111). Stimmt das Empfangsadressenzeichen weder mit der programmierten Adresse noch mit der "Rundruf"-Adresse überein, kehrt der Empfänger in Flagsuchzustand 130 zurück. Stimmt das Empfangsadressenzeichen entweder mit der programmierten Adresse oder mit der "Rundruf"-Adresse überein, oder wurde der Empfänger für Adressenerkennung nicht programmiert, erhöht der Empfänger seinen Zustand auf Datenzustand 133.

Im Datenzustand 133 nimmt der Empfänger den seriellen Empfangsdatenstrom vom Modemtreiber 60 an, führt bei Bedarf das Löschen der Nullbits aus, sammelt die Bits für ein komplettes Zeichen, und belegt den Empfangssilospeicher mit dem kompletten Zeichen zum Senden an UART 18. Der Empfänger erhöht seinen Zustand auf den Blockendenzustand 134, wenn er erkannt hat, daß

ein komplettes Zeichen ein Flagzeichen ist. Erkennt der Empfänger, daß der Empfangsdatenstrom vom Modemtreiber 60 sieben oder mehr logisch 1 Bits enthält, deutet der Empfänger dies als ein Abbruchzustand. schaltet auf den Flagsuchzustand 130 um, und setzt das Abbruchzustandsbit im Empfängerzustandsregister auf logisch 1.

Hat der Empfänger den Blockendenzustand 134 angenommen, wird das 16-Bit-Empfangs-FCS-Zeichen mit dem Inhalt des Empfangs-FCS-Generator verglichen. Der Empfangssilospeicher wird auch mit dem 16-Bit-Empfangs-FCS-Zeichen belegt. Darauf hin wird der Empfangssilospeicher mit dem Empfängerzustandwort einschließlich eines invertierten Paritätsbits für Übertragung an UART 18 belegt. Danach kehrt der Empfänger in den Flagsynchronisierungszustand 131 zurück.

Der Empfänger BSC ist dem Empfänger HDLC/SDLC gleich, mit Ausnahme: (1) kein Adressenzustand 132. so daß der BSC-Empfänger seinen Zustand direkt auf Datenzustand 133 erhöht; (2) kein Blockendenzustand 134, so daß der BSC-Empfänger im Datenzustand 133 verbleibt, bis er zurückgesetzt wird, worauf er erneut Flagsuchzustand 130 annimmt; (3) im Flagsuchzustand 130 sucht der BSC-Empfänger nach zwei Synchronisierungszeichen 110 und 111; und (4) der BSC-Empfänger erhöht seinen Zustand vom Flagsynchronisierungszustand 131 auf Datenzustand 133, wenn er erkennt, daß Nicht-Synchronisierungszeichen (Daten 112, CRC 113 oder PAD 114) den Synchronisierungszeichen 110 und 111 folgen.

### Ein-/Ausgangsregister

Im Asynchronbefehlszustand sendet externe Vorrichtungen 9 Anweisungen und liest Statusinformationen über sechszehn 8-Bit-Register. Die Bezeichnungen dieser Register sind: Senderhalterregister (THR), Empfänger-Pufferregister (RBR), Asynchronregister A (ARA), Asynchronregister B (ARB), Interrupt-Aktivierungsregister (IER), Interrupt-Erkennungsregister (IIR), Leitungssteuerregister (LCR), Modemsteuerregister (MCR), Leitungsstatusregister (LSR), BSC-Synchronisierungszeichenregister (BSCR), Modemstatusregister (MSR), HDLC/SDLC-Stationsadressenregister (SAR), Divisor-Latchregister

(DLL und DLM), Modemtreibergeschwindigkeitsregister (MERR), und Optionsregister (OR). Register THR ist ein Nur-Schreibe-Register. Register RBR und IIR sind Nur-Lese-Register. Register BSCR, SAR und OR werden nur für Synchronbetrieb verwendet.

Register THR, RBR, IER, IIR, LCR, MCR, LSR, DLL, DLM und MSR befinden sich in UART 18. Die übrigen Register befinden sich im Prozessor 47. Wie bereits erläutert, ist UART 18 so zu verstehen, daß einige der folgenden Registerbegriffe nur für Asynchronbetrieb, andere nur für Synchronbetrieb, und manche für beide Betriebsarten gelten.

Register THR ist der Eingangsregister des UART 18-Senders. Im Synchronbetrieb löst das Schreiben eines Zeichens in dieses Register eine Blocksendung aus. Das Zeichen wird erst dann gesendet, nachdem es in den Sender des Prozessors 47 übertragen worden ist und den Sendesilospeicher verlassen hat.

Register RBR ist der Ausgangsregister des UART 18-Empfängers. Bei jeder Ankunft eines Empfangszeichens im Register RBR wird ein Interrupt zur Signalisierung der Bereitstellung von Empfangsdaten erzeugt, es sei denn, daß dieses Interrupt durch Register IER maskiert wurde.

Register IER aktiviert/deaktiviert Interrupts unter den folgenden Bedingungen: Empfangsdaten bereit; Register THR leer; Modemstatuswechsel; und Leitungsstatuswechsel.

Register IIR ermöglicht die Prioritätsvergabe der Interrupts. In der bevorzugten Ausführungsform gibt es vier Interrupts-Prioritätsebenen: 1 - Leitungsstatus; 2 - Empfangsdaten bereit; 3 - Register THR leer; und 4 - Modemstatus. Die Leitungsstatusereignisse sind: Empfangsdaten im Register RBR bereit, Empfangsdaten mit Paritätsfehler behaftet, Überlauf des Registers RBR, Empfangsstopp-Interrupt, Register THR leer, Blockfehler, Empfangsblockendenfolge. FCS-Ergebnisprüfung, und UART 18 ohne Sendedaten für Prozessor 47. Die Modemstatusereignisse stellen eine Zustandsänderung einer der folgenden Signale dar: CTS auf Leitung 45, DCD auf Leitung 46, und RI auf Leitung 40. Register IIR, wenn adressiert, sichert das noch in Priorität höchstliegende Interrupt, und keine weiteren Interrupts werden quittiert, bis das Interrupt



mit der höchsten Priorität durch externe Vorrichtung 9 abgearbeitet worden ist.

Register LCR liefert die Anweisungen für Wortwahllänge, Stoppbitanzahl, Paritätswahl, Bitwahl der Divisorlatchadresse, Senderrücksetzen, Empfängerrücksetzen, Registeradressierung von UART 18 und HDLC/SDLC-Sendeabbruch. Eine Sender-Rücksetzanweisung setzt den Sender sofort in den Marke/Ruhe-Zustand 120 oder in den Flagruhezustand 121 zurück, je nachdem welcher Zustand im Register OR gewählt wurde. Diese Anweisung wird automatisch gelöscht, sobald der Vorgang zum Rücksetzen des Senders beendet ist. Eine Empfänger-Rücksetzanweisung setzt den Empfänger sofort in den Flagsuchzustand 130 zurück. Diese Anweisung wird automatisch gelöscht, sobald der Vorgang zum Rücksetzen des Empfängers beendet ist. Eine HDLC/SDLC-Sendeabbruchanweisung veranlaßt den Sender, das Abbruchzeichen sofort zu senden und den Marke/Ruhe-Zustand 120 oder den Flagruhezustand 121 aufzunehmen, je nachdem welcher Zustand im Register OR gewählt wurde. Diese Anweisung wird automatisch gelöscht, sobald das Abbruchzeichen gesendet wurde und der Sender den zugewiesenen Ruhezustand angenommen hat.

Register MCR steuert folgender Signale: das Signal DTR auf Leitung 36; das Signal RTS auf Leitung 37; das Signal OUT1 auf Leitung 32; das Signal OUT2 auf Leitung 27; und Rückschleifensignale von UART 18.

Register LSR zeigt die obengenannten Leitungstatusereignisse. Register BSCR enthält das 8-Bit-Synchronisationszeichen für BSC-Betrieb. Register MSR zeigt die obengenannten Modemstatusereignisse. Register SAR enthält die Stationsadresse für externe Vorrichtung 9 in HDLC/SDLC-Kommunikation.

Register DLL und DLM setzen die Datenrate für SIN und SOUT von UART 18. Diese Datenrate beträgt 9600 bps im Synchronbetrieb, und stellt die gewünschte Datenrate (150, 300, 600, usw. bps) im Asynchronbefehlsbetrieb und im Asynchrondatenbetrieb dar.

Register ARA und ARB setzen die Datenrate für P30 und P37. des Prozessors 47 entsprechend der Datenrate für SIN und

SOUT von UART 18. Diese Datenrate beträgt 9600 bps im Synchronbetrieb, und stellt die gewünschte Datenrate (150, 300, 600, usw. bps) im Asynchronbefehlsbetrieb und im Asynchrondatenbetrieb dar.

Register MERR setzt die Datenrate für P20 und P27 des Prozessors 47 sowie die Datenrate für TXD, RXD, CTXD und CRXD des Modemtreibers 60. Diese Datenrate ist die für Kommunikation mit entfernter Vorrichtung 100 erforderliche Datenrate.

Register OR wählt den Asynchron- oder Synchronbetrieb, wählt BSC- oder HDLC/SDLC-Kommunikation, aktiviert oder deaktiviert HDLC/SDLC-Adressenerkennung, wählt NRZ- oder NRZI-Codierung, und wählt Markenruhe oder Flag/Synchronruhe.

### **Steuerung und Überwachung**

Prozessor 47 steuert und überwacht den Betrieb des Modemtreibers 60 durch die Eingaben Lesen, NICHT-Schreiben und Steuern und die bidirektionellen Dateneingaben/Ausgaben (D0-D7) des Modemtreibers 60. Prozessor 47 steuert, zum Beispiel, die folgenden Betriebsparameter des Modemtreibers 60: Datenrate, automatische Antwortaktivierung, Analog-rückschleifen, Zeichenlängewahl, Zweiton-Mehrfachfrequenz (DTMF)-Wählen, Impulswählen, Schutzton-Aktivierung, Langleertrennzeichen empfangen, usw. Prozessor 47 überwacht auch, zum Beispiel, folgende Betriebsparameter des Modemtreibers 60: Wähldigitregister leer, Rufanzeige, Empfangsträger erfaßt, Tonerfassung usw.

Prozessor 47 muß externer Vorrichtung 9 von Änderungen beim Überwachungsbetrieb des Modemtreibers 60 in Kenntnis setzen. Prozessor 47 kann jedoch nur durch UART 18 mit externer Vorrichtung 9 kommunizieren. Daher werden sowohl die vom Prozessor 47 erzeugten Empfangsdatenworte als auch die Statusworte über UART 18 geleitet. Externe Vorrichtung 9 benötigt ein Verfahren, um erkennen zu können, ob es sich bei einem durch UART 18 auf Bus 10 gesetzten Wort, um ein Empfangsdatenwort oder ein Statuswort handelt.

Es wird daran erinnert, daß im Synchronbetrieb Prozessor 47 Start-, Stopp- und Paritäts-Bits zu den von ihm an UART 18

gesendeten Worten hinzufügt. Um Datenworte von Statusworten unterscheiden zu können, invertiert Prozessor 47 das Paritätsbit bei Statusworten. Das invertierte Paritätsbit veranlaßt UART 18 ein Interrupt zu erzeugen. Wenn externe Vorrichtung 9 arbeitet das Interrupt ab, liest sie das Statusregister von UART 18 und stellt fest, daß ein Paritätsfehler eingetreten ist. Dieser Paritätsfehler warnt externe Vorrichtung 9, daß das nächste Wort ein vom Prozessor 47 gesendetes Statuswort und nicht ein Empfangsdatenwort sein wird.

Im Synchronbetrieb werden sämtliche von externer Vorrichtung 9 an Prozessor 47 über UART 18 gesendeten Worte als Sendedatenworte und nicht als Anweisungen behandelt. Um also Anweisungen an Prozessor 47 senden zu können, sendet externe Vorrichtung 9 die Anweisung an UART 18, die NICHT-Leitung 36 für Datenterminal bereit (DTR) mit logisch 0 zu belegen. Ein logisch 0 auf Leitung 36 veranlaßt Prozessor 47, Synchronbetrieb zu verlassen und Asynchronbefehlsbetrieb aufzunehmen.

Sobald Prozessor 47 in Asynchronbefehlsbetrieb versetzt ist, behandelt er sämtliche Worte von UART 18 als Anweisungen von externer Vorrichtung 9, bis er eine Anweisung erhält, Synchronbetrieb oder Asynchrondatenbetrieb aufzunehmen.

Befindet sich Prozessor 47 im Asynchronbefehlzzustand, belegt er beide Leitungen 53 und 54 mit logisch 1. Dies veranlaßt Multiplexer 56, seinen Eingang X3 mit seinem Ausgang X und seinen Eingang Y3 mit seinem Ausgang Y zu verbinden. Da Eingang Y3 des Multiplexers 56 mit logisch 1 verbunden ist, wird Eingang CTXD des Modemtreibers 60 mit logisch 1 belegt, Dies veranlaßt Modemtreiber 60, den Marke/Ruhe-Zustand an entfernte Vorrichtung 100 zu senden.

Multiplexer 56 verbindet auch Ausgang CRXD des Modemtreibers 60 mit dem Empfangsdateneingang des Prozessors 47. In der bevorzugten Ausführungsform ignoriert Prozessor 47 die Empfangsdaten, wenn er sich im Asynchronbefehlsbetrieb befindet. Es können jedoch Anwendungen geben, in denen es wünschenswert ist, daß Prozessor 47 auf den Empfangsdatenstrom reagiert bzw. diesen weiterleitet, selbst wenn sich Prozessor 47 im Asynchronbefehlzzustand befindet.

Belegt Prozessor 47 Leitung 53 mit logisch 1 und Leitung 54 mit logisch 0, verbindet Multiplexer 56 seinen Eingang X2 mit seinem Ausgang X, und seinen Eingang Y2 mit seinem Ausgang Y. Hierdurch werden vom Modemtreiber 60 empfangene Daten direkt an UART 18, über Prozessor 47, und anschließend zum Modemtreiber geschickt. Diese Bedingung findet in der bevorzugten Ausführungsform keine Verwendung, sie ist jedoch vorgesehen, für den Fall, daß eine Anwendung vorkommen sollte, in der es wünschenswert wäre, daß Sendedaten auf irgendeine Weise durch Prozessor 47 verarbeitet werden, bevor sie an Modemtreiber 60 zur Übertragung an entfernte Vorrichtung 100 gesendet werden.

In manchen Anwendungen kann es wünschenswert sein, Anweisungen an Prozessor 47 senden zu können, ohne vorher Prozessor 47 veranlassen zu müssen, Synchronbetrieb zu verlassen. Zu diesem Zweck inspiziert Prozessor 47 die von UART 18 kommenden Sendeworte. Handelt es sich beim Sendewort um ein Zeichen "DLE", erfährt Prozessor 47 hierdurch, daß es sich beim darauffolgenden Wort um ein Anweisungswort, und nicht um ein zu sendendes Datenwort handelt. Wird jedoch eine Folge von zwei DLE-Zeichen empfangen, deutet Prozessor 47 diese als eine Anweisung, ein DLE-Zeichen zu senden, d.h. das erste DLE-Zeichen zu löschen und das zweite DLE-Zeichen an Modemtreiber 60 zu senden. Hierdurch besteht die Möglichkeit, ein DLE-Zeichen als Teil des Sendedatenstroms zu senden.

Ebenfalls, wenn eine bestimmte Anwendung verlangt, daß das UART 18 Interrupt nicht zur Kennzeichnung eines Statusworts verwendet wird, wird Prozessor 47 das DLE-Zeichen unmittelbar vor dem Statuswort in den an UART 18 gesendeten Empfangsdatenstrom einfügen. Wenn externe Vorrichtung 9 das DLE-Zeichen erfaßt, behandelt sie das anschließende Wort als ein Statuswort. Prozessor 47 überwacht auch den vom Modemtreiber 60 einkommenden Empfangsdatenstrom. Erfäßt Prozessor 47 ein DLE-Zeichen im Empfangsdatenstrom, schiebt er unmittelbar nach dem ersten DLE-Zeichen ein zweites DLE-Zeichen in den Empfangsdatenstrom zu UART 18 ein. Externe Vorrichtung 9 erkennt, daß zwei aufeinanderfolgende DLE-Zeichen einem DLE-Zeichen im Empfangsdatenstrom entspricht.

### Patentanspruch

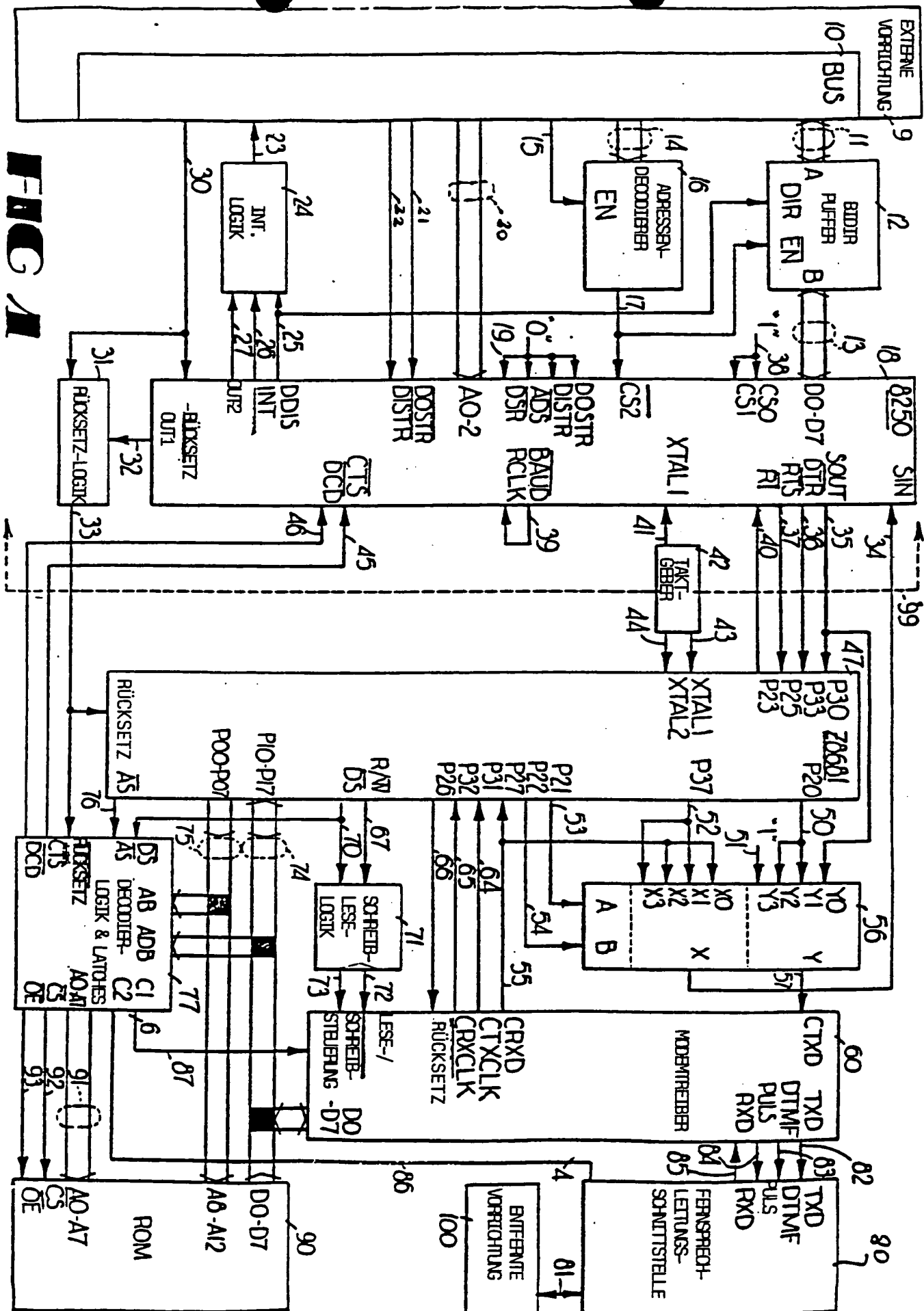
Verfahren zum Austauschen von Informationsdaten zwischen Vorrichtungen, indem die Informationsdaten zu einem Informationsdatenwort codiert werden, welches eine erste vorbestimmte Parität besitzt, und die Informationsdaten als eine Serie von Datenworten übertragen werden, welche Serie eine Vielzahl derartiger Informationsdatenworte umfaßt, sowie zum Austauschen von Steuerdaten zwischen den Vorrichtungen, gekennzeichnet durch:

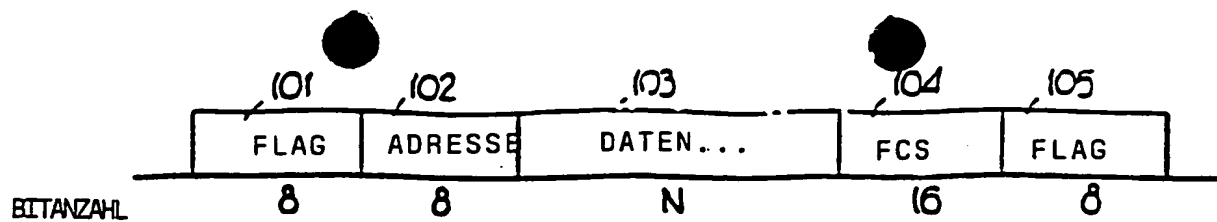
ein Übertragungsverfahren, wobei

- a) die Steuerdaten zu einem Steuerdatenwort codiert werden, welches eine zweite vorbestimmte Parität besitzt, und
- b) das Steuerdatenwort in die Serie von Datenworten eingefügt wird; sowie

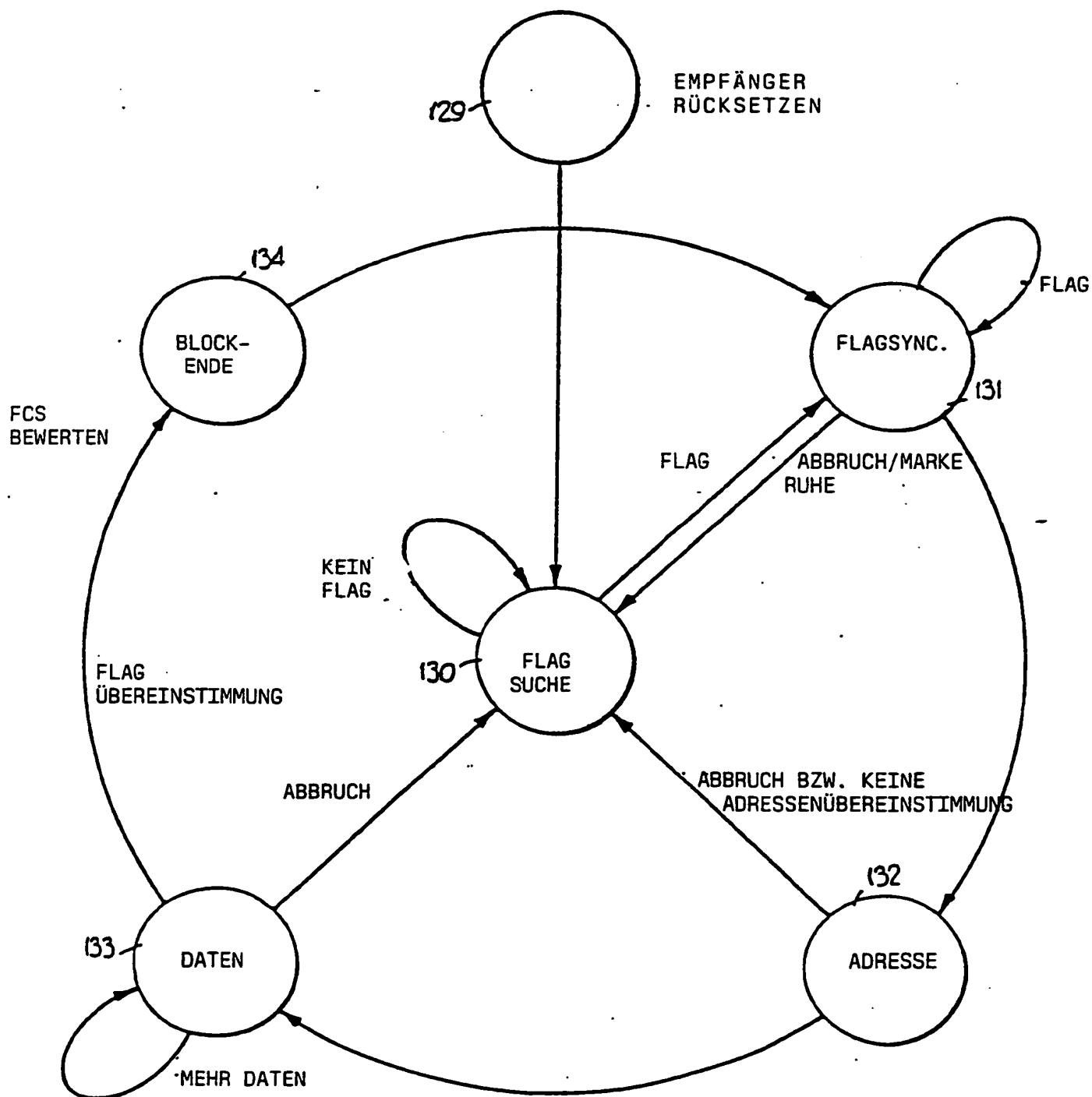
ein Empfangsverfahren, wobei

- c) die Parität jedes Datenwortes in der Serie von Datenworten geprüft wird und
- d) jedes Datenwort, welches die zweite vorbestimmte Parität besitzt, als eines der Steuerdatenworte weiterverarbeitet wird.

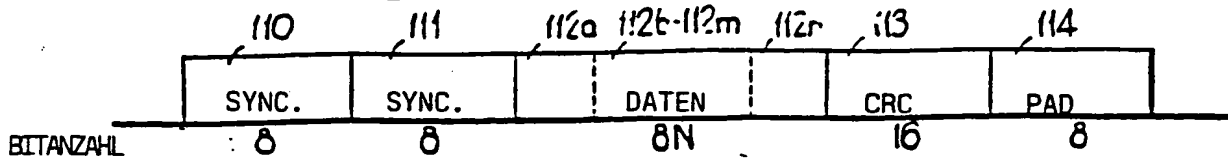




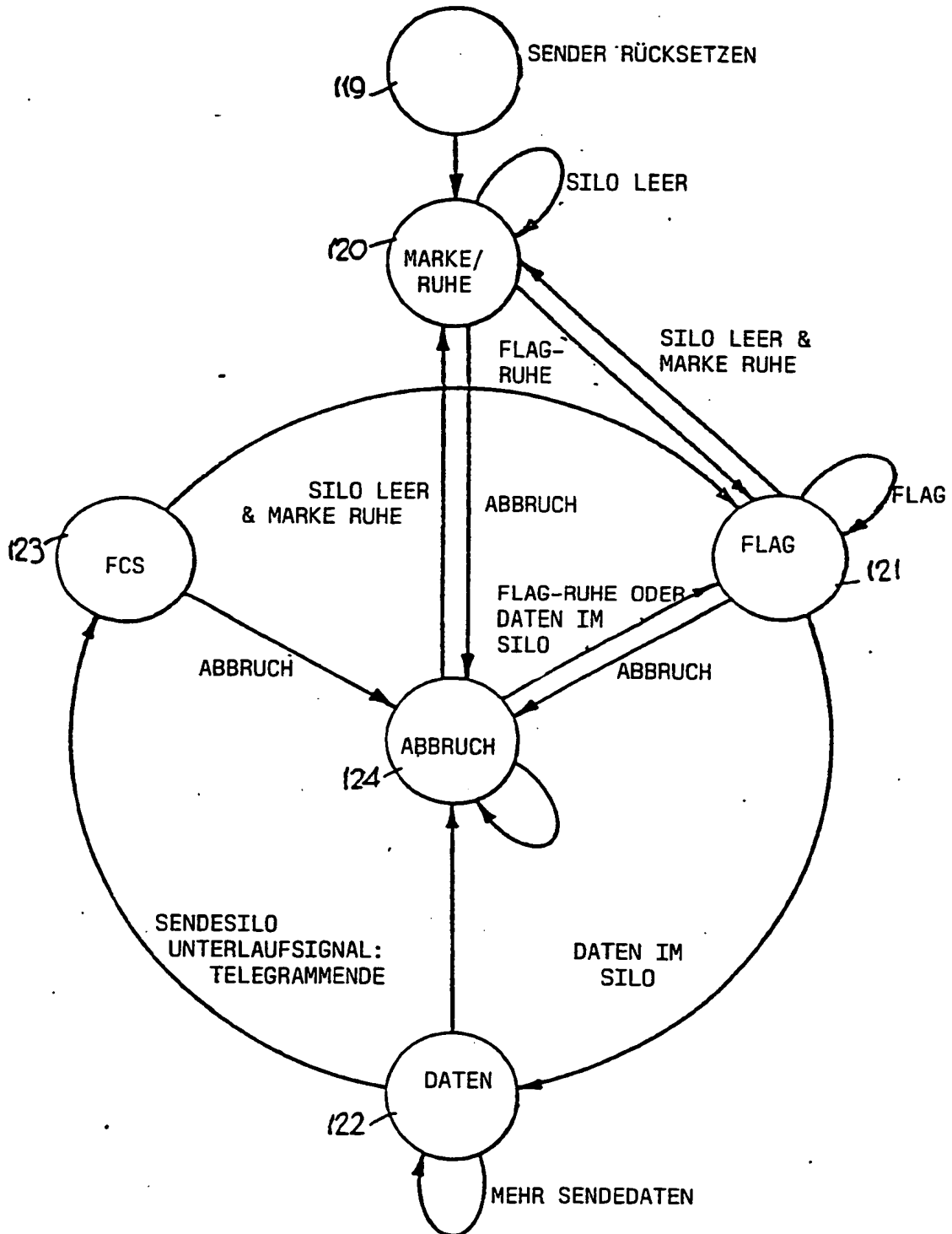
**FIG 2**



**FIG 5**



**FIG 3**



**FIG 4**